Premier Reference Source

Novel Design and Applications of Robotics Technologies

Dan Zhang and Bin Wei

IGI Global
DISSEMINATOR OF KNOWLEDGE

Novel Design and Applications of Robotics Technologies

Dan Zhang
York University, Canada

Bin Wei
York University, Canada

A volume in the Advances in
Computational Intelligence and
Robotics (ACIR) Book Series

IGI Global
DISSEMINATOR OF KNOWLEDGE

Published in the United States of America by
IGI Global
Engineering Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Names: Zhang, Dan, 1964- editor. | Wei, Bin, 1987- editor.
Title: Novel design and applications of robotics technologies / Dan Zhang and Bin Wei, editors.
Description: Hershey, PA : Engineering Science Reference, [2018] | Includes bibliographical references.
Identifiers: LCCN 2017038309| ISBN 9781522552765 (hardcover) | ISBN 9781522552772 (ebook)
Subjects: LCSH: Robotics--Technological innovations.
Classification: LCC TJ211 .N68 2018 | DDC 629.8/92--dc23 LC record available at https://lccn.loc.gov/2017038309

This book is published in the IGI Global book series Advances in Computational Intelligence and Robotics (ACIR) (ISSN: 2327-0411; eISSN: 2327-042X)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material.
The views expressed in this book are those of the authors, but not necessarily of the publisher.

For electronic access to this publication, please contact: eresources@igi-global.com.

Advances in Computational Intelligence and Robotics (ACIR) Book Series

ISSN:2327-0411
EISSN:2327-042X

Editor-in-Chief: Ivan Giannoccaro, University of Salento, Italy

MISSION

While intelligence is traditionally a term applied to humans and human cognition, technology has progressed in such a way to allow for the development of intelligent systems able to simulate many human traits. With this new era of simulated and artificial intelligence, much research is needed in order to continue to advance the field and also to evaluate the ethical and societal concerns of the existence of artificial life and machine learning.

The **Advances in Computational Intelligence and Robotics (ACIR) Book Series** encourages scholarly discourse on all topics pertaining to evolutionary computing, artificial life, computational intelligence, machine learning, and robotics. ACIR presents the latest research being conducted on diverse topics in intelligence technologies with the goal of advancing knowledge and applications in this rapidly evolving field.

COVERAGE

- Agent technologies
- Robotics
- Fuzzy systems
- Algorithmic Learning
- Brain Simulation
- Intelligent control
- Heuristics
- Evolutionary computing
- Pattern Recognition
- Artificial life

IGI Global is currently accepting manuscripts for publication within this series. To submit a proposal for a volume in this series, please contact our Acquisition Editors at Acquisitions@igi-global.com or visit: http://www.igi-global.com/publish/.

The Advances in Computational Intelligence and Robotics (ACIR) Book Series (ISSN 2327-0411) is published by IGI Global, 701 E. Chocolate Avenue, Hershey, PA 17033-1240, USA, www.igi-global.com. This series is composed of titles available for purchase individually; each title is edited to be contextually exclusive from any other title within the series. For pricing and ordering information please visit http://www.igi-global.com/book-series/advances-computational-intelligence-robotics/73674. Postmaster: Send all address changes to above address. ©© 2019 IGI Global. All rights, including translation in other languages reserved by the publisher. No part of this series may be reproduced or used in any form or by any means – graphics, electronic, or mechanical, including photocopying, recording, taping, or information and retrieval systems – without written permission from the publisher, except for non commercial, educational use, including classroom teaching purposes. The views expressed in this series are those of the authors, but not necessarily of IGI Global.

Titles in this Series

For a list of additional titles in this series, please visit:
https://www.igi-global.com/book-series/advances-computational-intelligence-robotics/73674

Handbook of Research on Predictive Modeling and Optimization Methods in Science ...
Dookie Kim (Kunsan National University, South Korea) Sanjiban Sekhar Roy (VIT University, India) Tim Länsivaara (Tampere University of Technology, Finland) Ravinesh Deo (University of Southern Queensland, Australia) and Pijush Samui (National Institute of Technology Patna, India)
Engineering Science Reference • ©2018 • 618pp • H/C (ISBN: 9781522547662) • US $355.00

Handbook of Research on Investigations in Artificial Life Research and Development
Maki Habib (The American University in Cairo, Egypt)
Engineering Science Reference • ©2018 • 501pp • H/C (ISBN: 9781522553960) • US $265.00

Critical Developments and Applications of Swarm Intelligence
Yuhui Shi (Southern University of Science and Technology, China)
Engineering Science Reference • ©2018 • 478pp • H/C (ISBN: 9781522551348) • US $235.00

Handbook of Research on Biomimetics and Biomedical Robotics
Maki Habib (The American University in Cairo, Egypt)
Engineering Science Reference • ©2018 • 532pp • H/C (ISBN: 9781522529934) • US $325.00

Androids, Cyborgs, and Robots in Contemporary Culture and Society
Steven John Thompson (University of Maryland University College, USA)
Engineering Science Reference • ©2018 • 286pp • H/C (ISBN: 9781522529736) • US $205.00

Developments and Trends in Intelligent Technologies and Smart Systems
Vijayan Sugumaran (Oakland University, USA)
Engineering Science Reference • ©2018 • 351pp • H/C (ISBN: 9781522536864) • US $215.00

Handbook of Research on Modeling, Analysis, and Application of Nature-Inspired ...
Sujata Dash (North Orissa University, India) B.K. Tripathy (VIT University, India) and Atta ur Rahman (University of Dammam, Saudi Arabia)
Engineering Science Reference • ©2018 • 538pp • H/C (ISBN: 9781522528579) • US $265.00

For an entire list of titles in this series, please visit:
https://www.igi-global.com/book-series/advances-computational-intelligence-robotics/73674

IGI Global
DISSEMINATOR OF KNOWLEDGE

701 East Chocolate Avenue, Hershey, PA 17033, USA
Tel: 717-533-8845 x100 • Fax: 717-533-8661
E-Mail: cust@igi-global.com • www.igi-global.com

Table of Contents

Detailed Table of Contents

In the last couple of decades, massive industrial application demands greatly accelerated the advances of control theories for robotic manipulator. However, the autonomous robotic capture is still facing many technical challenges. A novel visual servo kinematic control scheme for robotic manipulators to perform autonomous capture operation of a non-cooperative target is presented. An integrated algorithm of the photogrammetry and the adaptive extended Kalman filter is introduced to estimate the position and orientation of the target. In order to improve the reliability of the robotic control and to avoid the multiple solutions problem of inverse kinematics, a kinematics-based incremental control approach is adopted to control the robotic manipulator in real time. Validating experiments are performed on a custom built robotic manipulator with an eye-in-hand configuration. The experimental results demonstrate the effectiveness and robustness of the proposed control visual servo kinematic control scheme.

Presented in this chapter is a method for design and analysis of a mobile manipulator. The wrench induced by the movement of the robot arm will cause system tip-over or slip. In tip-over analysis, three cases are considered. The first case deals with the effect of the link weights and tip payload on the horizontal position of the CG. The second case deals with the effect of the joint speeds through the coupling terms including centrifugal forces and gyroscopic moments. The third case deals with the effect of the joint accelerations through the inertia forces and moments. In slip analysis, the first case considers the reaction force in relation to the stand-off distance between system and work-piece. The second and third cases investigate the effects of the joint speeds and accelerations. Then, the mobile platform is optimized to have maximum tip-over stability which optimizes the placement of the robot arm and accessory on the mobile platform. The effectiveness of the proposed method is demonstrated.

This chapter proposes a multi-agent organization model for self-adaptive software to examine the autonomous components and their self-adaptation that can be occurred at either the fine-grain behavior layer of a software agent or the coarse-grain organization layer of the roles that the agent plays. The authors design two-layer self-adaptation mechanisms and combine them with reinforcement learning together to tackle the uncertainty issues of self-adaptation, which enables software agents to make decisions on self-adaptation by learning at run-time to deal with various unanticipated changes. The reinforcement learning algorithms supporting fine-grain and coarse-grain adaptation mechanisms are designed. In order to support the development of self-adaptive software, the software architecture for individual agents, the development process and the software framework are proposed. A sample is developed in detail to illustrate our method and experiments are conducted to evaluate the effectiveness and efficiency of the proposed approach.

This chapter addresses the problem of coordinating the behavior of very large numbers of microrobots to assemble complex, hierarchically structured physical objects. The approach is patterned after morphogenetic processes during embryological development, in which masses of simple agents (cells) coordinate to produce complex three-dimensional structures. To ensure that the coordination mechanisms scale up to hundreds of thousands or millions of microrobots, the swarm is treated as a continuous mass using partial differential equations. A morphogenetic programming notation permits algorithms to be developed for coordinating dense masses of microrobots. The chapter presents algorithms and simulations for assembling segmented structures (artificial spines and legs) and for routing artificial neural fiber bundles. These algorithms scale over more than four orders of magnitude.

In this chapter, the authors propose an algorithm for the reduction of fault tree expressions that are generated from failure behavioral models. The significance of the sequencing of events is preserved during the generation and all along the reduction process, thus allowing full qualitative analysis. Thorough and detailed analysis results should positively impact the design of condition monitoring and failure prevention mechanisms. A behavioral model of a robotic system that exhibits sequence-dependent failures is used in the study.

The chapter gives an overview of the current developmental trends in service robotics. First the short history of service robots with its precursors is presented together with the definitions. The developmental trends and statistical data are summarized. The overview of service robots includes the ancient robot precursors, the middle ages, and the period of industrial revolution. The representative examples of different kinds of service robots built in the 20th and 21st centuries are also given. The chapter is concluded by discussing the perspectives of service robotics.

In recent years, many researchers have shown interest in developing search and rescue systems composed of one or multiple robots. To enhance the robotic systems, wireless sensor networks and internet of things (IoT) were integrated to give more awareness of the environments. Additionally, data exchanged in social media during emergency situations can help rescuers, decision makers, and the public to gain insight into the situation as it unfolds. In the first part of this chapter, the authors present a review of robotic system and their environments in search and rescue systems. Additionally, they explain the challenges related to these systems and the tasks that a robot or a multi-robot system should execute to fulfil the search and rescue activities. As a second part, the authors expose the systems that integrates WSNs and IoT with robots and the advantages that brings those. Furthermore, they expose and discuss the remarkable research, the challenges, and the open research challenges that include this cooperation.

Research and development of exoskeletons began as early as the 1960s. Recent advancement in technology has spurred a further research into the field specifically at rehabilitation and human performance augmentation. Human performance augmenting exoskeletons find use in the military, emergency services, industrial and space applications, and training. Rehabilitation exoskeletons assist in posture support and replacing lost function. Exoskeleton research is broadly broken up in this chapter by anthropometry: lower body, upper body, and extremities. The development for various anthropometry has their own unique set of challenges. This chapter provides a brief history, discusses current trends in research, looks at some of the technology involved in development, the potential benefits of using exoskeletons, and looks at the possible future improvements in research.

In artificial intelligence and robotics, one of the important issues is to design human interface. There are two issues: One is the machine-centered interaction design. Another one is the human-centered interaction design. This research aims at the latter issue. This chapter presents the interactive learning system to assist positive change in the preference of a human toward the true preference. Then evaluation of the awareness effect is discussed. The system behaves passively to reflect the human intelligence by visualizing the traces of his/her behaviors. Experimental results showed that subjects are divided into two groups, heavy users and light users, and that there are different effects between them under the same visualizing condition. They also showed that the authors' system improves the efficiency for deciding the most preferred plan for both heavy users and light users. As future research directions, a probabilistic event and its basic recommendation way are discussed.

This chapter concerns the problem of odor source localization by a team of mobile robots. A brief overview of odor source localization is given which is followed by related work. Three methods are proposed for odor source localization. These methods are largely inspired by gravitational search algorithm, grey wolf optimizer, and particle swarm optimization. Objective of the proposed approaches is to reduce the time required to localize the odor source by a team of mobile robots. The intensity of odor across the plume area is assumed to follow the Gaussian distribution. Robots start search from the corner of the workspace. As robots enter in the vicinity of plume area, they form groups using K-nearest neighbor algorithm. To avoid stagnation of the robots at local optima, search counter concept is used. Proposed approaches are tested and validated through simulation.

Preface

Robotics have been used in many areas, such as manufacturing, aerospace, medical, social services, and agriculture. The book *Novel Design and Applications of Robotics Technologies* comprises of enhanced, expanded, and updated versions of articles published in the 2013, 2014, 2015, and 2016 volume years of *International Journal of Robotics Applications and Technologies*. The overall theme of the book is to provide readers the state-of-the-art technologies in the novel design of robotic technologies and its real-world applications. Main topics in this book include design and applications of mobile manipulators, visual servo kinematic control for robotic manipulators, self-adaptations of multi-agent organizations using reinforcement learning, continuum mechanics for coordinating micro-robot swarms, history of service robots and its new trends, robotics in search and rescue activities, and passively interactive learning.

The book consists of 10 chapters. In Chapter 1, an incremental kinematic control strategy for the visual servo robotic manipulator with eye-in-hand configuration to perform autonomous capture of a non-cooperative target is presented. A vision based integrated algorithm of the photogrammetry and the AEKF is adopted for the kinematic state estimation of the target. Initialized by the photogrammetry and enhanced by solely adaptive process noise distribution of the Kalman filtering method, the motion of a non-cooperative target is accurately estimated by an eye-in-hand camera. Chapter 2 deals with the wheeled manipulator used for manufacturing, such as drilling, riveting or line drawing. The proposed analysis and optimization method is applied to design two mobile manipulators, one for drilling/riveting for aerospace manufacturing, another mobile manipulator is for baseline drawing for interior decoration. According to the optimization and analysis, in actual operation, the manipulator moves with constraints obtained as results from different case analysis and consequently no tip-over occurs. For slip issue, the path is planned with joint speeds/accelerations restrained to decrease the slip phenomenon and ensure the accuracy of localization. Details can be referred to Chapter 2. Chapter 3 proposes a multi-agent organization model for self-adaptive software to examine the autonomous components and their self-adaptation that can be occurred at either the fine-grain behavior layer of a software agent or the coarse-grain organization layer of the roles that the agent plays. The authors design two-layer self-adaptation mechanisms, and combine them with reinforcement learning together to tackle the uncertainty issues of self-adaptation, which enables software agents to make decisions on self-adaptation by learning at run-time to deal with various unanticipated changes. The reinforcement learning algorithms supporting fine-grain and coarse-grain adaptation mechanisms are designed. In order to support the development of self-adaptive software, the software architecture for individual agents, the development process and the software framework are proposed. A sample is developed in detail to illustrate our method and experiments are conducted to evaluate the effectiveness and efficiency of the proposed approach. Chapter 4 addresses the problem of coordinating the behavior of very large numbers of microrobots to assemble complex, hierarchically structured physical objects. The approach is patterned after morphogenetic processes during embryological development, in which masses of simple agents (cells) coordinate to produce complex three-dimensional structures. To ensure that the coordination mechanisms scale up to hundreds of thousands or millions of microrobots, the swarm is treated as a continuous mass using partial differential equations. A morphogenetic programming notation permits algorithms to be developed for coordinating dense masses of microrobots. The chapter presents algorithms and simulations for assembling segmented structures (artificial spines and legs) and for routing artificial neural fiber bundles. In Chapter 5, the authors outlined a lack of suitable techniques that generate, synthesize and analyze fault trees with the significant sequencing of events from models specific to robotics systems. Most of the existing work deals with static fault trees; however, less efforts are on generation of dynamic fault trees but with a main focus on quantitative analysis. In this regard, this chapter puts emphasis on such important issue by highlighting effects of sequence-dependent failures and the importance of an adequate qualitative analysis that reduces some of the difficulties encountered in probabilistic assessments. The second contribution is a solution proposed for the reduction of fault trees with more expressive power and that are generated from automata descriptions of failure behaviors. In several safety-critical domains, failure behavioral models are increasingly automata based like with EAST-ADL in the automotive industry, Altarica and AADL in aerospace, and RobotML in the robotic domain. Therefore, the algorithms proposed in this chapter complement the previous contributions regarding advanced generation and synthesis of such fault tree expressions. These allow improved failure prediction capabilities which should be used as an input to the design of proactive failure avoidance measures. Likewise, as redundancy allocation is a popular reliability improvement technique, the authors also presented the usefulness of this work as part of an approach to cost-effectively improving dependability of the overall robotic system via a process of multi-objective optimization of design models. Chapter 6 presents the history of service robots and new trends. It is pointed out that the further development of robot intelligence is associated with cloud robotics, freeing robots from their computational limits and thus increasing their capabilities. This will enable the development of sensor rich robots with high adaptability and autonomy. The deep learning algorithms will allow the robots to access and process efficiently the data available in the media, by that increasing their perception abilities, what will result in significant enhancement of their "understanding." The robots will learn faster and more efficiently. Knowledge sharing between robots is another new trend requiring a common ontology. Very different robots having different sensors and different effectors will share their "knowledge," thus improving their learning processes. In Chapter 7, the authors first reviewed robotic search and rescue systems, by defining these latter and their environments. Additionally, the authors presented challenges related to these latter and tasks that a robot or a multi-robot system should execute to fulfil the search and rescue activities. Then, the authors exposed a system that integrates WSN with robotic system in order to better accomplish rescue tasks. The advantages that bring the WSN to robotic systems in search and rescue activities are presented. Finally, the authors cited tasks and missions that are achieved in a better way with a cooperation of WSN and robots. Recent research shows interest to the use of a cooperation of wireless sensor networks and robots for search and rescue due to the reliability and strength that give WSNs to robotic systems, which can replace the human presence. Since this field is recently adopted to the search and rescue systems, there is many areas, which continue to be studied. For example, solutions that propose to save energy of the radio by sampling, processing and aggregating data, if one adopts them to search and rescue activities, it could be a big step forward. This gives long life batteries for both robots and sensors, which avoids a redeployment of sensors and the change of batteries of robots on the middle of the search and rescue activity due to holes on the coverage and to avoid the risk of losing the tracked robots or victims. Chapter 8 provides a brief history of exoskeleton development from a human factors and ergonomic intervention standpoint. It discusses the current research with respect to the innate human-machine interface and the incorporation of exoskeletons for ergonomic intervention. Some novel exoskeletons based on their anatomical categories of lower body, upper body, extremities (hands/feet), and full body exoskeletons will be discussed. The chapter concludes by covering the benefits of exoskeletons in rehabilitation, industrial applications, and military applications, as well as discuss some of the issues faced when designing exoskeletons. In Chapter 9, the authors describe awareness-based recommendation by passively interactive learning system. For considering preference shift of a user, the authors proposed user-centered interactive recommendation by visualizing both the recommendation space with prepared recommendation plans and the user's preference trace as the history of the recommendation in it. Experimental results showed that subjects are divided into two groups, heavy users and light users, and that there are different effects between heavy users and light users under the same visualizing condition. Heavy users reduced the number of reference and the total reference time. Chapter 10 discusses an approach which minimizes the time required to locate the odor source, by utilizing the strengths of GSA, GWO and PSO. The proposed method is the fusion of above three. GWO has been used in concatenation with the hybrid of GSA and PSO. First candidate solutions are created by the hybrid of GSA and PSO, and then resultant solutions are fed into GWO as initial solutions to get the final solutions and vice-versa.

The editors would like to acknowledge the support from Jan Travers and Mariah Gilbert of IGI Global. We also would like to express our appreciation to all the authors for their contributions to this book and make this book possible.

Dan Zhang
York University, Canada

Bin Wei
York University, Canada

Chapter 1
Visual Servo Kinematic Control for Robotic Manipulators

Zheng Hong Zhu
York University, Canada

Gangqi Dong
Northwestern Polytechnical University, China

ABSTRACT

In the last couple of decades, massive industrial application demands greatly accelerated the advances of control theories for robotic manipulator. However, the autonomous robotic capture is still facing many technical challenges. A novel visual servo kinematic control scheme for robotic manipulators to perform autonomous capture operation of a non-cooperative target is presented. An integrated algorithm of the photogrammetry and the adaptive extended Kalman filter is introduced to estimate the position and orientation of the target. In order to improve the reliability of the robotic control and to avoid the multiple solutions problem of inverse kinematics, a kinematics-based incremental control approach is adopted to control the robotic manipulator in real time. Validating experiments are performed on a custom built robotic manipulator with an eye-in-hand configuration. The experimental results demonstrate the effectiveness and robustness of the proposed control visual servo kinematic control scheme.

DOI: 10.4018/978-1-5225-5276-5.ch001

INTRODUCTION

Due to increasing requirements of efficiency, dexterity and autonomy, robotic manipulators have been extensively employed in industrial and other applications to perform tasks that are monotonous, complex, dangerous, or even impossible to be conducted by human beings (Yoshida, 2009). Researchers have dedicated considerable efforts on design, modeling and control of robotic manipulators to accomplish many different kinds of missions, where autonomous capturing is considered as one of the key functions and with great engineering potentials not only in industrial applications but also in space missions (Larouche et al, 2014; Huang et al, 2016; Yu et al, 2016; Zhang et al, 2017).

The task of autonomous robotic capture of non-cooperative targets is not trivial. Although numerous enabling techniques have been proposed, autonomous robotic capture is still very challenging. Especially in space applications, the autonomous capture related techniques attract more attention than others. A preliminary concept design of guidance, navigation and control architecture was proposed in (Jankovic et al, 2015) for a safe and fuel-efficient robotic capture of a non-cooperative target in active debris removal missions, where the capture process was divided into three phases: orbital rendezvous, proximity maneuver and robotic capture. Since the autonomous orbital rendezvous and proximity maneuver are mature technologies and have been successfully performed in many other space missions, researchers mainly focus on the final autonomous capture stage of a non-cooperative target by space robotic manipulators. A vision-based prediction and motion-planning scheme was introduced in (Aghili et al, 2012) for robotic capturing of free-floating tumbling objects with uncertain dynamics. An attitude takeover control approach was proposed in (Huang et al, 2016) to stabilize the reconfigurable spacecraft in post-capture of target by space robotic manipulators. However, according to the recent study by Flores-Abad et al. in (Flores-Abad et al, 2014), the up to date successful space missions with the participation of robotic manipulator, such as active debris removal and on-orbit servicing, mainly relied on the human-in-the-loop control to gain higher reliability. In fact, the control problem of space robotic manipulators with free floating base is somewhat equivalent to ground ones with fixed base, enabled by generalized Jacobian matrix (Umetani et la, 2010), virtual manipulator (Vafa et al, 1987) and dynamically equivalent manipulator (Liang et al, 1998) technologies. In order to focus on the robotic control problem, the presented approach is performed on a fixed-base manipulator to prove the concept of the presented control scheme. Additional efforts, such as replacing the Jacobian matrix in the current kinematic approach with a generalized Jacobian matrix, or mapping the kinematics of a free floating manipulator to a fixed-base manipulator, may be required to apply the presented approach to free floating base space robotic manipulators.

Two challenges arise sequentially in the autonomous capture of non-cooperative targets by robotic manipulators. First, the robotic manipulator has to promptly know where the target is, which means the pose and motion of the target has to be estimated in real time. Since vision system is generally favored in navigation and control during rendezvous and docking operations (Sabatini, 2015; Gasbarri et al, 2014) as well as in robotic control (Chaumette et al, 2006; Chamuette et al, 2007) for monitoring, detecting and tracking purpose due to its intuitive and non-contact natures, it is employed to detect, estimate and model the target in this work. Generally, existing vision-based target pose and motion estimation methods could be classified into four categories, such as geometric methods, learning-based optimization methods, offline methods and filtering-based methods (Dong et al, 2016; Martinez et al, 2011; Konolige et al, 2011; Janabi-Sharifi et al, 2010; Caron et al, 2014; Huang et al, 2015). Geometric methods are prone to image noises since they are based on the current measurement only. Learning-based optimization methods require sufficient knowledge of the target in advance, which is not available if the target is unknown. The offline methods perform estimation afterwards and is not suitable for real-time applications. Therefore, filtering-based estimation methods, especially the Kalman filter based methods, are generally favored in the literature as they do not require a priori knowledge of the targets (Chen, 2012). Depending on the camera configuration, the vision systems are classified into two types: the eye-in-hand and the eye-to-hand. As the name implies, the camera is mounted on the robotic manipulator in the eye-in-hand configuration while detached from the manipulator and fixed in workspace in the eye-to-hand configuration. Consequently, the eye-to-hand camera monitors the whole workspace to provide global yet less accurate estimates of target's position and motion in a global frame. In contrast, the eye-in-hand camera is usually mounted close to the end-effector and provides a close and more accurate scene of the target in a local frame. The accuracy of the estimates of position and motion increases as the end-effector approaches the target. Accordingly, the eye-in-hand configuration is adopted in the current study. To avoid the loss of target tracking caused by the moving target and the motion of the eye-in-hand camera, an integrated estimation algorithm of photogrammetry and adaptive extended Kalman filter (Dong et al, 2016) is adopted to estimate the position and motion of the unknown target in real time.

The second challenge is how to determine the trajectory of the end-effector so as to perform autonomous capture by the robotic manipulator. Most of the existing robotic control methods were based on dynamics, where the control input is usually joint actuating force or torque, and for redundant manipulators, multiple solutions problem in the inverse kinematics of robotic manipulators will inevitably encountered due to the periodicity of trigonometric functions and the redundant geometric configuration. Since the loading capacity is no longer a concerned key issue, especially in agravic space environment, and the positioning accuracy is higher than force/torque control, kinematics-based joint-position-level control has great potential in autonomous robotic control. Therefore, a kinematic control approach is introduced to accomplish the autonomous robotic capture. The incremental nature of the presented approach assures the avoidance of the multiple solutions problem. The stability of the control scheme is analyzed and the theoretical proof is provided. The control scheme is also validated experimentally on a custom built robotic manipulator with an eye-in-hand configuration. The experimental results demonstrated the effectiveness and robustness of the vision-based kinematic control scheme for autonomous robotic capture.

Vision-Based Target Estimation

Consider a non-cooperative target and a vision-based robotic manipulator system. Without loss of generality, assume the global frame, denoted by $\mathcal{G}$, is fixed in the inertia space; the camera frame, denoted by $\mathcal{C}$, is attached to the center of the image plane of camera and the target frame, denoted by $\mathcal{T}$, is fixed to the target body, respectively. Further assume the position of a feature point on the target is known with respect to (w.r.t) $\mathcal{T}$ and is denoted by $\left\{x_T, y_T, z_T\right\}^T$ in the target frame. Then, the position of this feature point w.r.t $\mathcal{C}$, denoted by $\left\{x_C, y_C, z_C\right\}^T$ in the camera frame, can be derived by an augmented homogeneous transformation from $\mathcal{T}$ to $\mathcal{C}$, such that

$$\begin{Bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{Bmatrix} = \begin{bmatrix} & & & x_{To} \\ & \mathbf{R}_{TC} & & y_{To} \\ & & & z_{To} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{Bmatrix} \tag{1}$$

where $\left\{x_{To}, y_{To}, z_{To}\right\}^T$ denotes the origin position of $\mathcal{T}$ w.r.t $\mathcal{C}$ in the target frame and $\mathbf{R}_{TC}$ stands for the transformation or rotational matrix from $\mathcal{T}$ to $\mathcal{C}$. The transformation matrix is formed by the trigonometric functions of the Euler angles between axes of $\mathcal{T}$ and $\mathcal{C}$, denoted by $\left\{\theta_x, \theta_y, \theta_z\right\}^T$.

From Eq. 1, $\left\{x_C, y_C, z_C\right\}^T$ can be expressed as the functions of $\left\{x_{To}, y_{To}, z_{To}\right\}^T$ and $\left\{\theta_x, \theta_y, \theta_z\right\}^T$. Note that $\left\{x_{To}, y_{To}, z_{To}\right\}^T$ and $\left\{\theta_x, \theta_y, \theta_z\right\}^T$ are the unknown position and orientation of $\mathcal{T}$ w.r.t $\mathcal{C}$ and will be determined by the photogrammetry as follows. Let a feature point on the target is projected onto the image plane by a pinhole camera, such that

$$\begin{Bmatrix} x_m \\ z_m \end{Bmatrix} = -\frac{f}{y_C - f} \begin{Bmatrix} x_C \\ z_C \end{Bmatrix} \tag{2}$$

where f is the focal length of the camera and $\left\{x_m, \mathrm{z}_m\right\}^T$ is the projected coordinates of the feature point on the image plane and can be obtained from the image processing. Here, the y-axis of $\mathcal{C}$ is assumed to be perpendicular with the image plane and pointing to the target.

Rearranging Eq. 2 yields two independent equations with six unknowns for one feature point

$$\begin{cases} x_m y_C - x_m f + x_C f = 0 \\ z_m y_C - z_m f + z_C f = 0 \end{cases} \tag{3}$$

Theoretically, a minimum of three distinguished feature points are required to solve for six unknowns, but ambiguous solutions may occur due to the periodic solutions of trigonometric functions. To avoid the ambiguity, at least four feature points are usually adopted to increase the robustness of solution (Larouche et al, 2014). The resulting equations are highly nonlinear and are solved iteratively by the least square approach with an initial guess. Due to the process is solely based on the current measurement, the memoryless characteristics of the photogrammetry leads to the solution sensitive to the measurement noises. Furthermore, the computational cost may subject to the errors between initial guess and the real solution because the initial guess is made arbitrarily, leading to the large variation of the system sampling time. The latter is not desirable in real-time applications. In addition, similar to the limitation of other geometric approach, the photogrammetry does not solve for the target motion directly. In order to address these challenges, an integrated algorithm of photogrammetry and adaptive extended Kalman filter (AEKF) is employed to estimate the kinematic states of the target timely.

The state variable vector of a target is defined based on its kinematic variables, such as

$$\mathbf{X} = \left\{x_{To}, \dot{x}_{To}, y_{To}, \dot{y}_{To}, z_{To}, \dot{z}_{To}, \theta_x, \dot{\theta}_x, \theta_y, \dot{\theta}_y, \theta_z, \dot{\theta}_z\right\}^T \tag{4}$$

The overhead dot denotes the time derivatives of the corresponding variables. By assuming the target moving at a constant speed within a sufficient small sampling time interval t_s and the acceleration vector, denoted by $\omega = \left\{\ddot{x}_{To}, \ddot{y}_{To}, \ddot{z}_{To}, \ddot{\theta}_x, \ddot{\theta}_y, \ddot{\theta}_z\right\}^T$, as the process noise for simplicity, the system model is defined as

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{B}\omega_{k-1} \tag{5}$$

The subscripts k and $k-1$ in Eq. 5 indicate the current and previous sampling time steps. The matrices $\mathbf{A}$ and $\mathbf{B}$ can be written as

$$\mathbf{A} = diag\begin{bmatrix}\mathbf{a} & \mathbf{a} & \mathbf{a} & \mathbf{a} & \mathbf{a} & \mathbf{a}\end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix}1 & t_s \\ 0 & 1\end{bmatrix} \tag{6}$$

$$\mathbf{B} = diag\begin{bmatrix}\mathbf{b} & \mathbf{b} & \mathbf{b} & \mathbf{b} & \mathbf{b} & \mathbf{b}\end{bmatrix}, \quad \mathbf{b} = \begin{Bmatrix}t_s^2/2 \\ t_s\end{Bmatrix} \tag{7}$$

In order to apply the Kalman filtering method, the process noise ω is assumed to obey the Gaussian distribution with normal mean vector $\mathbf{q}$ and covariance matrix $\mathbf{Q}$, such that

$$\omega \sim \mathcal{N}\left(\mathbf{q}, \mathbf{Q}\right) \tag{8}$$

Because the pose of the target are detected by the camera, based on the pinhole camera model described by Eq. 2, the measurement model is defined as

$$\mathbf{Z}_k = \mathbf{h}\left(\mathbf{X}_k\right) + \mu_k \tag{9}$$

where $\mathbf{Z}$ is the position vector of the feature points in image plane, μ stands for the measurement noise of the camera, and $\mathbf{h}\left(\mathbf{X}\right)$ is defined by

$$\mathbf{h}\left(\mathbf{X}\right) = -\frac{f}{y_C - f}\begin{Bmatrix}x_C \\ z_C\end{Bmatrix} \tag{10}$$

Note that the measurement noise of the camera depends on operation conditions, such as illumination, environmental temperature, jittering of pixels, etc. It is reasonable to assume that the measurement noise obeys the zero mean Gaussian distribution with covariance matrix $\mathbf{R}$, such that,

$$\mu \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}\right) \tag{11}$$

Generally, the process and measurement noises are time-varying and it is desirable to adaptively update them for high accuracy. However, it is difficult to distinguish them from each other in reality. As a result, adaptively updating the distributions of both process and measurement noises may not be robust (Song et al, 2008). Considering the fact that the measurement noises are characteristics of the measurement system and are independent on the target, it is reasonable to assume the process noise and the measurement noise are independent to each other. Thus, the covariance matrix of the measurement noises $\mathbf{R}$ can be determined in advance experimentally, while only $\mathbf{q}$ and $\mathbf{Q}$ are adaptively updated for the process noises in the current work.

However, challenge arises for the determination of $\mathbf{q}$ and $\mathbf{Q}$ due to the non-cooperative nature of the target as well as the eye-in-hand configuration of the vision system where the camera is affected by the motion of the end-effector. For given initial conditions, the state variable vector $\mathbf{X}$ and its corresponding covariance matrix $\mathbf{P}$ in the next step can be predicted by following the typical process of Kalman filtering algorithm, such that,

$$\mathbf{X}_{k|k-1} = \mathbf{A}\mathbf{X}_{k-1|k-1} + \mathbf{B}\mathbf{q}_{k-1} \tag{12}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^{T} + \mathbf{B}\mathbf{Q}_{k-1}\mathbf{B}^{T} \tag{13}$$

Then, the Kalman gain is derived by

$$\mathbf{K}_{g} = \mathbf{P}_{k|k-1}\mathbf{H}_{k}^{T}\left(\mathbf{H}_{k}\mathbf{P}_{k|k-1}\mathbf{H}_{k}^{T} + \mathbf{R}\right)^{-1} \tag{14}$$

where $\mathbf{H}$ denotes the Jacobian matrix of the measurement model, which is defined by

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}\left(\mathbf{X}\right)}{\partial \mathbf{X}}\right|_{\mathbf{X}=\mathbf{X}_{k|k-1}} \tag{15}$$

Once the measurement vector $\mathbf{Z}$ is obtained from the image processing, the state variable vector and its corresponding covariance matrix in the next step are updated by

$$\mathbf{X}_{k|k} = \mathbf{X}_{k|k-1} + \mathbf{K}_g\left(\mathbf{Z}_k - \mathbf{h}\left(\mathbf{X}_{k|k-1}\right)\right) \tag{16}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_g\mathbf{H}_k\mathbf{P}_{k|k-1} \tag{17}$$

In order to adaptively update the normal mean and covariance matrix of the process noise, an intuitive approximation of $\mathbf{q}$ at the $j\text{-th}$ step is defined as,

$$\hat{\mathbf{q}}_j = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\left(\mathbf{X}_{j|j} - \mathbf{A}\mathbf{X}_{j-1|j-1}\right) \tag{18}$$

Accordingly, the unbiased estimates of $\mathbf{q}$ and $\mathbf{Q}$ at the time k can be evaluated under the assumption that the process noises are independent and uniformly distributed over N time steps, such that,

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \frac{1}{N}\left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N}\right) \tag{19}$$

$$\mathbf{Q}_k = \mathbf{Q}_{k-1} + \frac{1}{N-1}\left(\begin{array}{l}\left(\hat{\mathbf{q}}_k - \mathbf{q}_k\right)\left(\hat{\mathbf{q}}_k - \mathbf{q}_k\right)^T - \left(\hat{\mathbf{q}}_{k-N} - \mathbf{q}_k\right)\left(\hat{\mathbf{q}}_{k-N} - \mathbf{q}_k\right)^T \\ + \frac{1}{N}\left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N}\right)\left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N}\right)^T + \frac{N-1}{N}\left(\Lambda_{k-N} - \Lambda_k\right)\end{array}\right) \tag{20}$$

$$\mathbf{\Lambda}_k = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\left(\mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T - \mathbf{P}_{k|k}\right)\mathbf{B}\left(\mathbf{B}^T\mathbf{B}\right)^{-1} \tag{21}$$

The unbiased estimates of $\mathbf{q}$ and $\mathbf{Q}$ at the time k as described by Eqs. 19, 20 and 21 are the adaptive distribution updates of the process noises. Since the initial state variables of the non-cooperative target is unknown, a poor initial guess would lead to a long convergence time of the AEKF, which is not desired in real-time application. Thus, we adopted the estimation results of photogrammetry to initialize the AEKF to accelerate the convergence of the AEKF. Eqs. 12-21 are iteratively performed and the estimate of the state variable vector is obtained by Eq. 16 during each iteration.

It should be noted that the estimated position and motion of the target are w.r.t $\mathcal{C}$. For a concise description, let ${}^{C}\mathbf{X}_T, {}^{C}\dot{\mathbf{X}}_T$ and $\mathbf{X}_T, \dot{\mathbf{X}}_T$ denote the estimated position and motion of the target w.r.t $\mathcal{C}$ and $\mathcal{G}$, respectively. According to the geometric configuration of the camera, the transformation from $\mathcal{C}$ to $\mathcal{G}$ can be easily accomplished by the homogeneous transformation matrix, denoted by $\mathbf{T}$, such that

$$\begin{Bmatrix} \mathbf{X}_T \\ 1 \end{Bmatrix} = \mathbf{T} \begin{Bmatrix} {}^{C}\mathbf{X}_T \\ 1 \end{Bmatrix} \tag{22}$$

$$\begin{Bmatrix} \dot{\mathbf{X}}_T \\ 0 \end{Bmatrix} = \mathbf{T} \begin{Bmatrix} {}^{C}\dot{\mathbf{X}}_T \\ 0 \end{Bmatrix} + \dot{\mathbf{T}} \begin{Bmatrix} {}^{C}\mathbf{X}_T \\ 1 \end{Bmatrix} \tag{23}$$

Based on the above discussion, the estimates of target's position and motion w.r.t $\mathcal{G}$ are obtained by the integrated photogrammetry and AEKF algorithm.

Incremental Kinematic Control

The kinematics of a robotic manipulator defines the kinematic relationship between the end-effector in Cartesian space and the robotic manipulator in joint space. For given joint angle vector and angular velocity vector of the actuators, denoted by Θ and $\dot{\Theta}$ respectively, the position and velocity of the end-effector w.r.t $\mathcal{G}$, denoted by $\mathbf{X}_E$ and $\dot{\mathbf{X}}_E$ respectively, can be easily obtained by the so called forward kinematics, such as

$$\mathbf{X}_E = \mathbf{f}\left(\Theta\right) \tag{24}$$

$$\dot{\mathbf{X}}_E = \mathbf{J}\dot{\Theta} \tag{25}$$

where $\Theta \in \mathbf{R}^n$, $\mathbf{X}_E \in \mathbf{R}^m$ and $\mathbf{J}$ represents the Jacobian matrix of the robotic manipulator and is derived by the partial derivative of the right hand side of Eq. 24 w.r.t Θ, such as, $\mathbf{J} = \partial\mathbf{f}/\partial\Theta$.

In contrast, for the given position and velocity of the end-effector w.r.t $\mathcal{G}$, the inverse kinematics is required to solve for the corresponding joint angle and angular velocity of the manipulator. The inverse of Eq. may induce multiple solutions due to the periodicity of trigonometric functions and the redundant geometric configuration of the robotic manipulator. By neglecting the null space solution, the inverse of Eq. 25 can be written as

$$\dot{\Theta} = \mathbf{J}^{\dagger}\dot{\mathbf{X}}_E \tag{26}$$

where $\mathbf{J}^{\dagger}$ denotes the pseudo inverse of the Jacobian matrix $\mathbf{J}$.

It is usually assumed that the Jacobian matrix $\mathbf{J}$ is full row rank. Thus, the pseudo inverse may be simplified to the right inverse, such that,

$$\mathbf{J}^{\dagger} = \mathbf{J}^T\left(\mathbf{J}\mathbf{J}^T\right)^{-1} \tag{27}$$

We assume the target is always reachable by the robotic manipulator and the speed of the end-effector is always greater than the speed of the target in order to ensure the superior maneuverability of the robotic manipulator over the target. Based on its current position and velocity, the future position of the target w.r.t $\mathcal{G}$, denoted by $\mathbf{X}_{T'}$, can be predicted as

$$\mathbf{X}_{T'} = \mathbf{X}_T + \dot{\mathbf{X}}_T t_s \tag{28}$$

In order to guide the end-effector moving directly towards the predicted target position in the next moment, the instantaneous velocity of the end-effector in the Cartesian space should have the same direction with the vector defined by $\mathbf{X}_{T'} - \mathbf{X}_E$, such that,

$$\dot{\mathbf{X}}_E = \lambda\mathbf{n}_{ET'} \tag{29}$$

where λ is the positive scale factor of the instantaneous velocity of the end-effector and $\mathbf{n}_{ET'}$ denotes the unit vector of $\mathbf{X}_{T'} - \mathbf{X}_{E}$, such that

$$\mathbf{n}_{ET'} = \frac{\mathbf{X}_{T'} - \mathbf{X}_{E}}{\left|\mathbf{X}_{T'} - \mathbf{X}_{E}\right|} \tag{30}$$

Note that λ is unknown and should be determined. Substituting Eq. 29 into Eq. 26 leads to

$$\dot{\Theta} = \lambda \mathbf{J}^{\dagger} \mathbf{n}_{ET'} \tag{31}$$

Define $\dot{\theta}_{i}$, p_{ij} and $n_{ET'j}$ as the elements of $\dot{\Theta}$, $\mathbf{J}^{\dagger}$ and $\mathbf{n}_{ET'}$, respectively. Then, Eq. can be decomposed into n scalar equations, such that,

$$\dot{\theta}_{i} = \lambda \sum_{j=1}^{m} p_{ij} n_{ET'j}, \quad i = 1, 2, \cdots, n. \tag{32}$$

Assume the angular velocities of joint actuators in both forward and reversal directions are limited by $\dot{\Theta}_{\max} = \left\{\dot{\theta}_{\max 1}, \dot{\theta}_{\max 2}, \cdots, \dot{\theta}_{\max n}\right\}^{T}$. Applying the box constraint $-\dot{\theta}_{\max i}, \leq \theta_{i} \leq \dot{\theta}_{\max i}$ to the right hand side of Eq. 36 yields

$$-\dot{\theta}_{\max i} \leq \lambda \sum_{j=1}^{m} p_{ij} n_{ET'j} \leq \dot{\theta}_{\max i}, \quad i = 1, 2, \cdots, n. \tag{33}$$

Solving the inequalities in Eq. 35 independently for λ yields n sets of possible solutions, each denoted by λ_{i}. The upper bound of intersection of n sets of λ_{i} defines the desired scale factor of the instantaneous velocity of the end-effector,

$$\lambda = \max\left\{\lambda_{1} \cap \lambda_{2} \cap \cdots \cap \lambda_{n}\right\} \tag{34}$$

Substituting Eq. 34 into Eq. 31 yields the instantaneous angular velocity vector of the joint actuators, such as

$$\dot{\Theta} = \lambda \mathbf{J}^{\dagger} \mathbf{n}_{ET'} \tag{35}$$

Accordingly, the incremental angular position control input of the joint actuators in the next time step, which drives the end-effector towards the target directly, is obtained as

$$\delta\Theta = \dot{\Theta} t_s \tag{36}$$

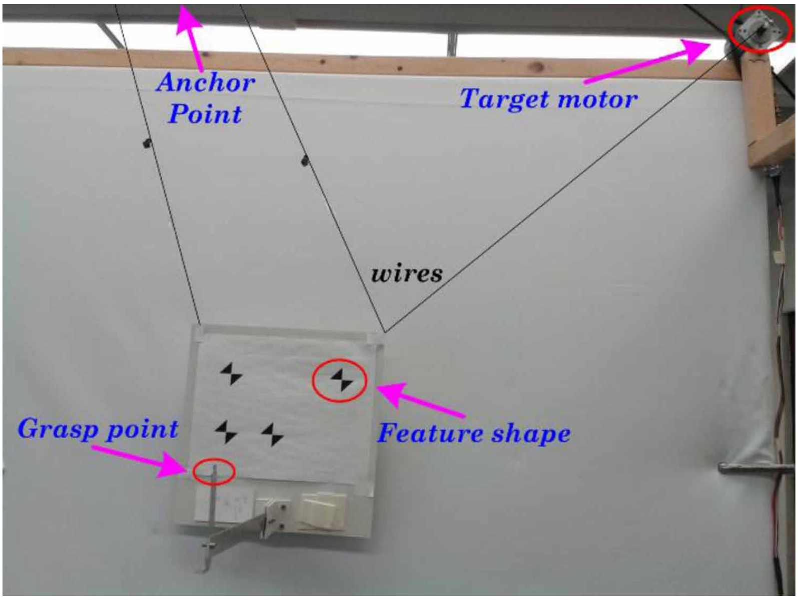
By applying the control input vector in Eq. 36 to the joint actuators, the end-effector moves towards the target by an increment at each time interval in the Cartesian space. The procedure defined by Eq. 22-36 is iteratively performed until the position error between the end-effector and the target is within a pre-defined tolerance and the capture action will be taken at the end of the procedure.

EXPERIMENTAL VALIDATION

The proposed integrated motion estimation algorithm and incremental kinematic control strategy are validated experimentally on an independent target system and a custom built robotic manipulator with an eye-in-hand configuration. As shown in Figure 1, the independent target system is suspended by fishing wires, anchored on the ceiling at the top left corner and driven by a single stepper motor fixed at the top right corner. By operating the target system independently to the robot controller, a non-cooperative target is generated since the target motion is unknown to the robotic manipulator and there is no communication between the target and the robotic manipulator. Four carefully designed low-noise feature shapes are adopted in order to improve the accuracy and efficiency of the image processing. In addition, the target is set to move 50 sampling steps (around 2s) ahead of the activation of the robotic manipulator to simulate a dynamic target. The tolerance for the tracking error is set as (0.01m, 0.01m, 0.01m).By detecting and grouping the corners of each feature shape, the centers of feature shapes are calculated and tracked for the kinematic state estimation of the non-cooperative target.

The experimental setup of the custom built robotic manipulator with an eye-in-hand configuration is shown in Figure 2. The motion of the end-effector is controlled by three revolute joints, namely torso, shoulder and elbow, which are denoted by θ_1, θ_2 and θ_3, respectively. It should be noted that the degrees of freedom θ_4 and θ_5 are designated for grasping the target and are not used in the current control law.

Figure 1. Experimental setup of the non-cooperative target system

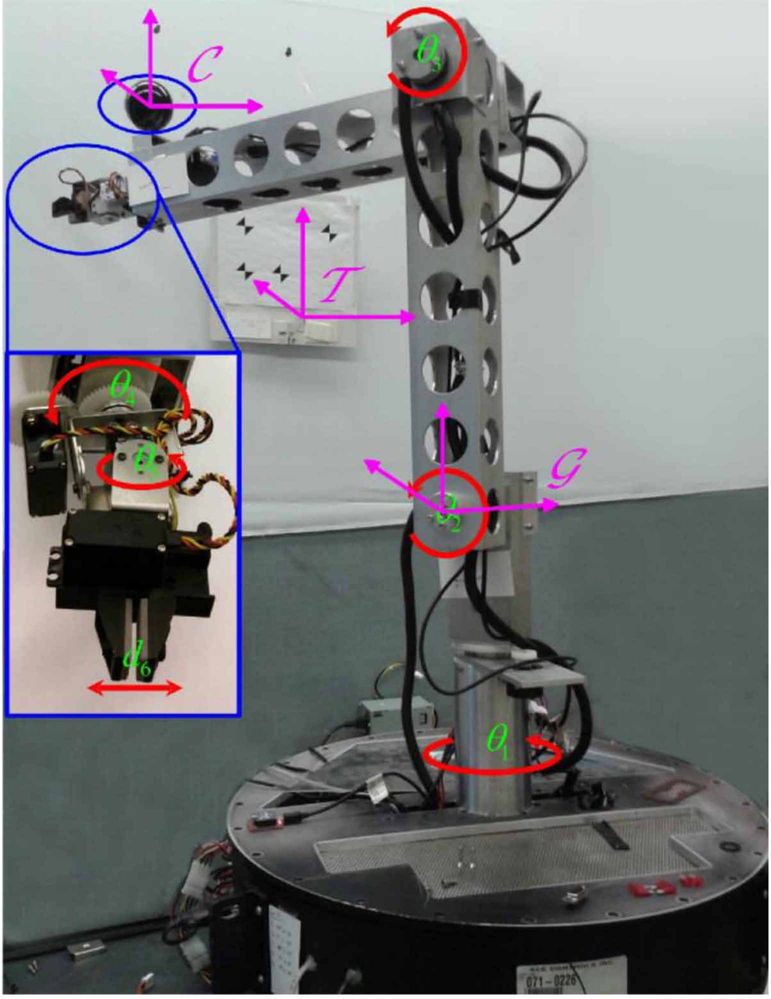


Shaft encoders were employed to measure joint angles in real time. Home position of the robotic manipulator is set to $\theta_1 = 0^\circ,\ \theta_2 = 90^\circ,\ \theta_3 = 0^\circ$. The link length between the shoulder and elbow joints is 0.4540 m and the link length between the elbow and the end-effector is 0.4445 m, as shown in Figure 2.

The experimental results are shown in Figures 3 - 7. Figure 3 shows an autonomous robotic capture process of a non-cooperative target. Initially, the robotic manipulator was at the home position. The red rectangle indicates the region of interest for the image processing at the beginning. After the target is locked, the optical flow function of the OpenCV library takes over the tracking task of the four features on the target. The target was activated 50 sampling time steps (roughly two seconds) ahead of the robotic manipulator in order to generate a dynamic non-cooperative target. The capture was deemed to successfully achieve when the distance between the end-effector and the target was reduced within the predefined tolerance.

To analyze the capture process in detail, Figure 4 illustrates the estimates of target's motion by the eye-in-hand camera w.r.t $\mathcal{C}$. During the experiment, the target motor was independently programmed to move the target within the workspace of the robotic manipulator. As mentioned above, the robotic manipulator was activated 50 sampling time steps after the activation of the target. This is reflected in Figure 4 where the estimated target motion is quite smooth because the camera was stationary. Once the robotic manipulator was activated, the eye-in-hand camera was no longer

Figure 2. Custom-built robotic manipulator with eye-in-hand configuration



stationary and the estimates of the target's motion were coupled with the motion of the robotic manipulator and become oscillation. This is much obviously shown in the velocity estimates.

Once the target motion is estimated, according to Eqs. -, the incremental control inputs for the joint actuators are obtained as shown in Figure 5. Joint angles of the robotic manipulator are measured by the shaft encoders directly and fed back to the controller in real time, as shown in Figure 6. As can be seen, the robotic manipulator behaves almost uniformly in joint space, except the torso angle θ_1, which is affected by the flexible coupling between the torso actuator and the first link. This kind of

Figure 3. Autonomous robotic capture process of the non-cooperative target

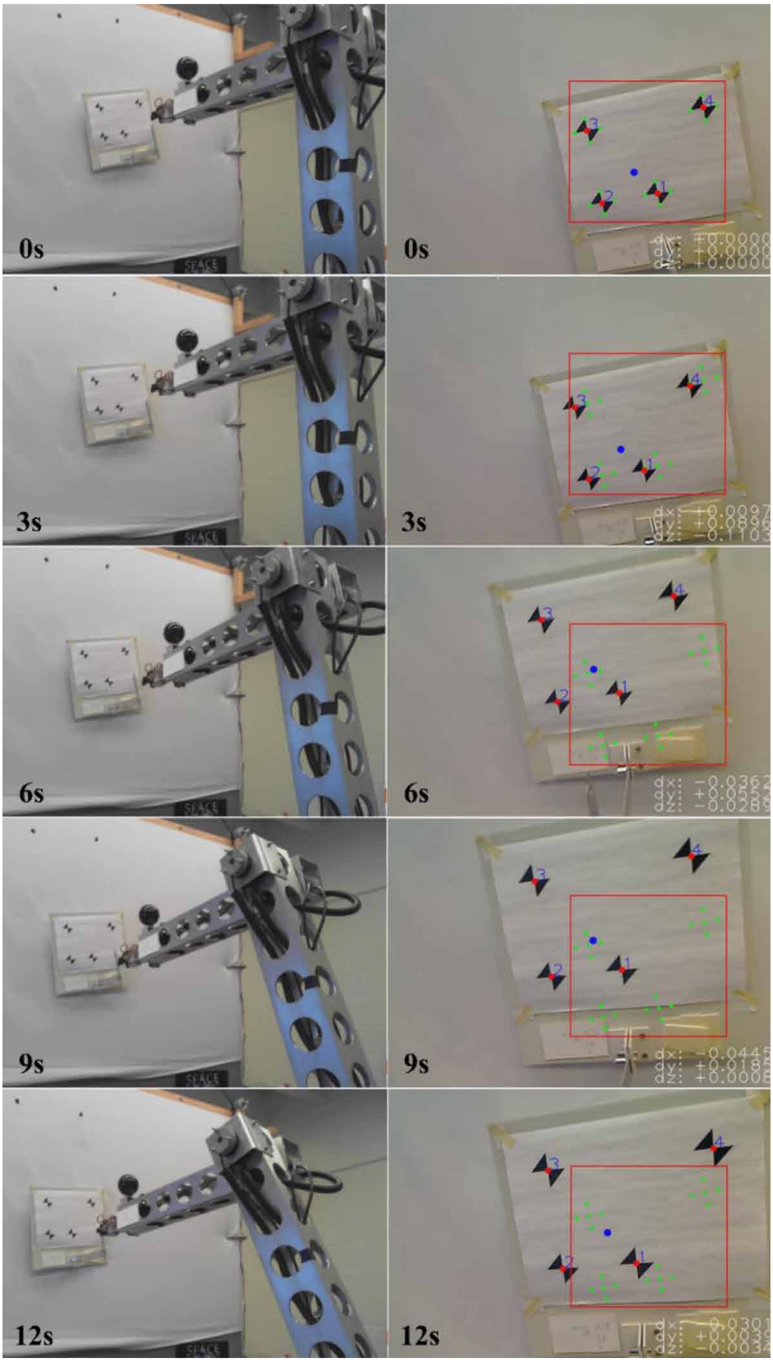


Figure 4. Estimation results of the target motion w.r.t $\mathcal{C}$

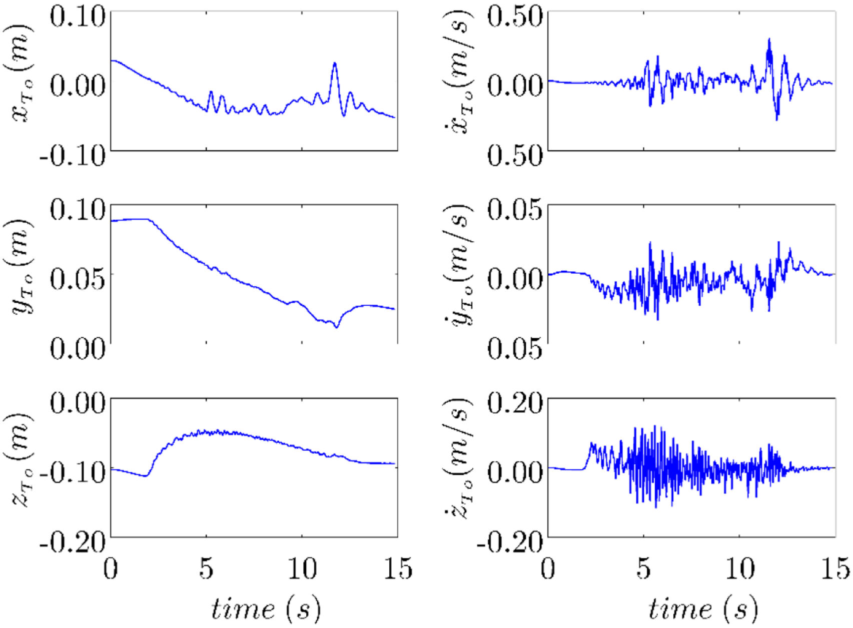


Figure 5. Time history of incremental control input of joint actuators

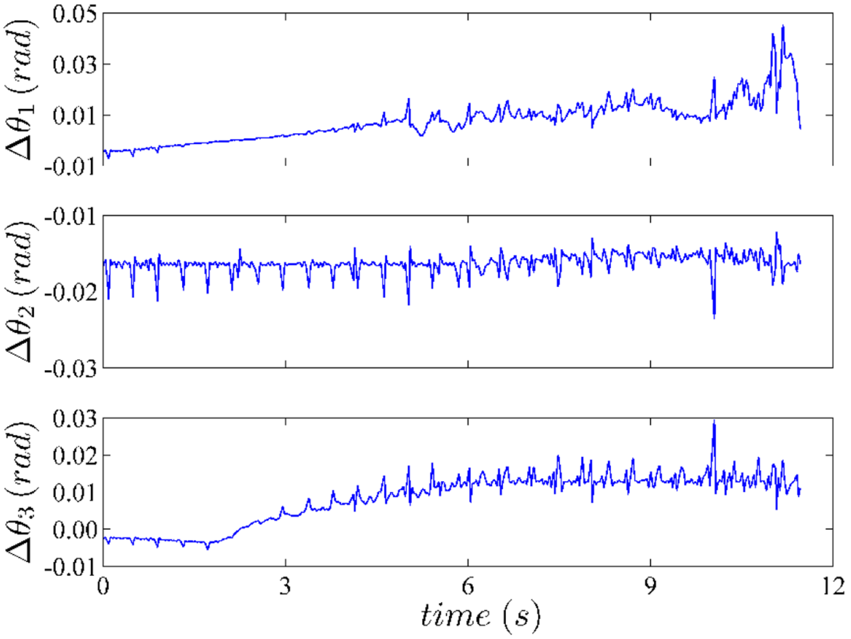
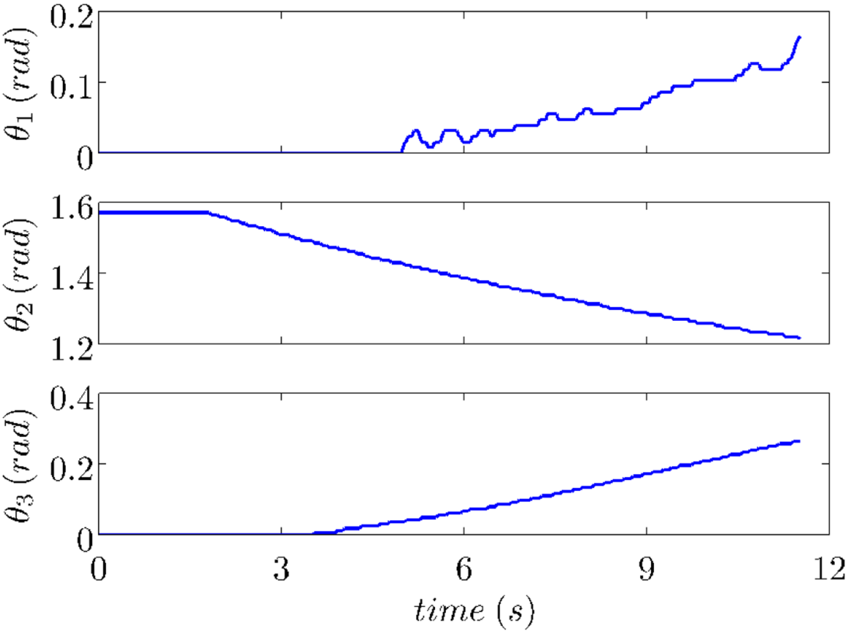


Figure 6. Time history of joint angles measured by the shaft encoders



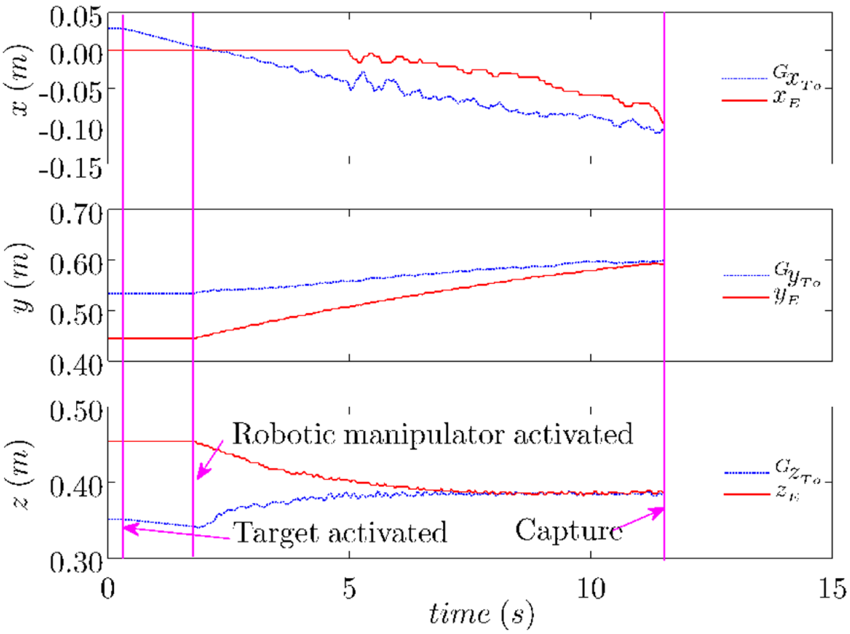
mechanical design is to ease the alignment requirement of rotational axes. Nevertheless, it shows the control strategy is robust to handle the disturbance not considered in the control law.

Figure 7 describes the time histories of the positions of the end-effector and the target. Since the position of the end-effector is w.r.t $\mathcal{G}$, in order to compare the position of the end-effector with the target, the target position is transformed to $\mathcal{G}$ by Eq., represented by ${}^{G}x_{To}$, ${}^{G}y_{To}$ and ${}^{G}z_{To}$ in the legends. It can be seen that, after the robotic manipulator was activated, the end-effector approached and captured the target autonomously at around 12 seconds successfully. The experimental results demonstrate the effectiveness and robustness of the estimation algorithm for the non-cooperative target as well as the kinematics based control strategy for autonomous robotic capture.

CONCLUSION

An incremental kinematic control strategy for the visual servo robotic manipulator with eye-in-hand configuration to perform autonomous capture of a non-cooperative target is presented. A vision based integrated algorithm of the photogrammetry and

Figure 7. Time histories of the positions of the end-effector and the target, w.r.t $\mathcal{G}$



the AEKF is adopted for the kinematic state estimation of the target. Initialized by the photogrammetry and enhanced by solely adaptive process noise distribution of the Kalman filtering method, the motion of a non-cooperative target is accurately estimated by an eye-in-hand camera. In order to avoid the multiple solutions problem of the inverse kinematics in robotics, incremental joint control input is generated by the kinematics based control. Since the end-effector is always aiming to the target position in the next moment, the possibility of losing target tracking by the eye-in-hand camera is reduced substantially. Validating experiments have been performed by a custom robotic manipulator. The test results of a successful capture demonstrate the effectiveness of the proposed visual servo kinematic control scheme.

ACKNOWLEDGMENT

This work is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fundamental Research Funds for the Central Universities (Grant No. G2017KY0307).

REFERENCES

Aghili, F. (2012). A Prediction and Motion-Planning Scheme for Visually Guided Robotic Capturing of Free-Floating Tumbling Objects With Uncertain Dynamics. *IEEE T Robot*, *28*(3), 634–649. doi:10.1109/TRO.2011.2179581

Caron, G., Dame, A., & Marchand, E. (2014). Direct model based visual tracking and pose estimation using mutual information. *Image and Vision Computing*, *32*(1), 54–63. doi:10.1016/j.imavis.2013.10.007

Chaumette, F., & Hutchinson, S. (2006). Visual servo control. I. Basic approaches, IEEE Robotics &amp;amp;amp. *Automation Magazine*, *13*(4), 82–90. doi:10.1109/MRA.2006.250573

Chaumette, F., & Hutchinson, S. (2007). Visual servo control - Part II: Advanced approaches. *IEEE Robotics & Automation Magazine*, *14*(1), 109–118. doi:10.1109/MRA.2007.339609

Chen, S. Y. (2012). Kalman Filter for Robot Vision: A Survey. *Ieee T Ind Electron*, *59*(11), 4409–4420. doi:10.1109/TIE.2011.2162714

Dong, G. Q., & Zhu, Z. H. (2016). Autonomous robotic capture of non-cooperative target by adaptive extended Kalman filter based visual servo. *Acta Astronautica*, *122*, 209–218. doi:10.1016/j.actaastro.2016.02.003

Flores-Abad, A., Ma, O., Pham, K., & Ulrich, S. (2014). A review of space robotics technologies for on-orbit servicing. *Progress in Aerospace Sciences*, *68*, 1–26. doi:10.1016/j.paerosci.2014.03.002

Gasbarri, P., Sabatini, M., & Palmerini, G. B. (2014). Ground tests for vision based determination and control of formation flying spacecraft trajectories. *Acta Astronautica*, *102*, 378–391. doi:10.1016/j.actaastro.2013.11.035

Huang, P., Wang, D., Meng, Z., Zhang, F., & Guo, J. (2015). Adaptive Postcapture Backstepping Control for Tumbling Tethered Space Robot–Target Combination. *Journal of Guidance, Control, and Dynamics*, *39*, 1–7.

Huang, P. F., Wang, M., Meng, Z. J., Zhang, F., Liu, Z. X., & Chang, H. T. (2016). Reconfigurable spacecraft attitude takeover control in post-capture of target by space manipulators. *Journal of the Franklin Institute*, *353*(9), 1985–2008. doi:10.1016/j.jfranklin.2016.03.011

Huang, P. F., Zhang, F., Meng, Z. J., & Liu, Z. X. (2016). Adaptive control for space debris removal with uncertain kinematics, dynamics and states. *Acta Astronautica*, *128*, 416–430. doi:10.1016/j.actaastro.2016.07.043

Janabi-Sharifi, F., & Marey, M. (2010). A Kalman-Filter-Based Method for Pose Estimation in Visual Servoing. *Ieee T Robot*, *26*(5), 939–947. doi:10.1109/TRO.2010.2061290

Jankovic, M., Paul, J., & Kirchner, F. (2015). GNC architecture for autonomous robotic capture of a non-cooperative target: Preliminary concept design. *Advances in Space Research*, *57*(8), 1715–1736. doi:10.1016/j.asr.2015.05.018

Konolige, K., Agrawal, M., & Sola, J. (2011). Large-scale visual odometry for rough terrain. In *Robotics Research* (pp. 201–212). Springer.

Larouche, B. P., & Zhu, Z. H. (2014). Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control. *Autonomous Robots*, *37*(2), 157–167. doi:10.100710514-014-9383-2

Liang, B., Xu, Y. S., & Bergerman, M. (1998). Mapping a space manipulator to a dynamically equivalent manipulator. *J Dyn Syst-T Asme*, *120*(1), 1–7. doi:10.1115/1.2801316

Martínez, C., Mondragón, I. F., Olivares-Méndez, M. A., & Campoy, P. (2011). On-board and ground visual pose estimation techniques for UAV control. *Journal of Intelligent & Robotic Systems*, *61*(1-4), 301–320. doi:10.100710846-010-9505-9

Sabatini, M., Palmerini, G. B., & Gasbarri, P. (2015). A testbed for visual based navigation and control during space rendezvous operations. *Acta Astronautica*, *117*, 184–196. doi:10.1016/j.actaastro.2015.07.026

Song, Q., & Han, J.-D. (2008). An adaptive UKF algorithm for the state and parameter estimations of a mobile robot. *Acta Automatica Sinica*, *34*(1), 72–79. doi:10.3724/SP.J.1004.2008.00072

Umetani, Y., & Yoshida, K. (2010). Resolved motion rate control of space manipulators with generalized Jacobian matrix. *Robotics & Automation IEEE Transactions on*, *5*(3), 303–314. doi:10.1109/70.34766

Vafa, Z., & Dubowsky, S. (1987). On the dynamics of manipulators in space using the virtual manipulator approach. *IEEE International Conference on Robotics and Automation. Proceedings*, 579-585. 10.1109/ROBOT.1987.1088032

Yoshida, K. (2009). Achievements in Space Robotics Expanding the Horizons of Service and Exploration. *IEEE Robot Autom Mag, 16*, 20-28.

Yu, Z. W., Liu, X. F., & Cai, G. P. (2016). Dynamics modeling and control of a 6-DOF space robot with flexible panels for capturing a free floating target. *Acta Astronautica*, *128*, 560–572. doi:10.1016/j.actaastro.2016.08.012

Zhang, B., Liang, B., Wang, Z. W., Mi, Y. L., Zhang, Y. M., & Chen, Z. (2017). Coordinated stabilization for space robot after capturing a noncooperative target with large inertia. *Acta Astronautica*, *134*, 75–84. doi:10.1016/j.actaastro.2017.01.041

APPENDIX

Stability Analysis of the Predictive Kinematic Controller

Define the tracking error of the robotic capture as

$$\mathbf{E} = \mathbf{X}_T - \mathbf{X}_E \tag{37}$$

$$\dot{\mathbf{E}} = \dot{\mathbf{X}}_T - \dot{\mathbf{X}}_E \tag{38}$$

Further define the Lyapunov function as

$$\mathbf{V} = \frac{1}{2}\mathbf{E}^T\mathbf{E} \tag{39}$$

Then take the first order time derivative of Eq. leads to

$$\dot{\mathbf{V}} = \mathbf{E}^T\dot{\mathbf{E}} \tag{40}$$

The geometry of the approach is shown in Figure 8. The angle between the vector direction of $\mathbf{E} = \mathbf{X}_T - \mathbf{X}_E$ and the vector direction of $\dot{\mathbf{X}}_T$ is defined as φ, and the angle between the vector direction of $\mathbf{E} = \mathbf{X}_T - \mathbf{X}_E$ and the vector direction of $\mathbf{X}_{T'} - \mathbf{X}_E$ is defined as ϕ.

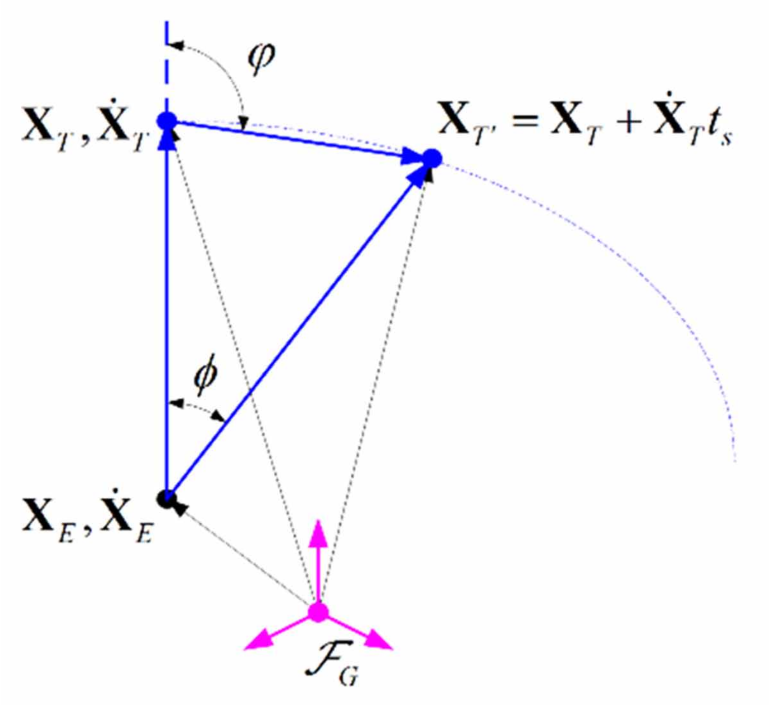
It should be noted that φ is an exterior angle of a triangle and ϕ is an interior angle of the triangle. Thus, with the consideration that the sampling time t_s is sufficient small, the following relationships keep

$$0 < \phi < \pi/2 \tag{41}$$

$$0 < \varphi < \pi \tag{42}$$

$$\phi < \varphi \tag{43}$$

Figure 8. Geometry of the predictive kinematic control


φ
$\mathbf{X}_T, \dot{\mathbf{X}}_T$
$\mathbf{X}_{T'} = \mathbf{X}_T + \dot{\mathbf{X}}_T t_s$
ϕ
$\mathbf{X}_E, \dot{\mathbf{X}}_E$
$\mathcal{F}_G$

According to the monotone decreasing property of cosine functions in the interval $\left[0, \pi\right]$ we obtain

$$\cos\varphi < \cos\phi \tag{44}$$

$$\cos\phi > 0 \tag{45}$$

From Eq. 29 and Eq. 30 we have

$$\dot{\mathbf{X}}_E = \frac{\lambda}{\left|\mathbf{X}_{T'} - \mathbf{X}_E\right|}\left(\mathbf{X}_{T'} - \mathbf{X}_E\right) \tag{46}$$

Substitute Eq. 46 into Eq. 38 yields

$$\dot{\mathbf{E}} = \dot{\mathbf{X}}_T - \frac{\lambda}{\left|\mathbf{X}_{T'} - \mathbf{X}_E\right|}\left(\mathbf{X}_{T'} - \mathbf{X}_E\right) \tag{47}$$

Further substitute Eq. 47 into Eq. 40 leads to

$$\dot{\mathbf{V}} = \mathbf{E}^T\dot{\mathbf{E}} = \mathbf{E}^T\dot{\mathbf{X}}_T - \frac{\lambda}{\left|\mathbf{X}_{T'} - \mathbf{X}_E\right|}\mathbf{E}^T\left(\mathbf{X}_{T'} - \mathbf{X}_E\right) \tag{48}$$

According to the definition of dot product, we have

$$\mathbf{E}^T\dot{\mathbf{X}}_T = \left|\mathbf{E}\right|\left|\dot{\mathbf{X}}_T\right|\cos\varphi \tag{49}$$

$$\mathbf{E}^T\left(\mathbf{X}_{T'} - \mathbf{X}_E\right) = \left|\mathbf{E}\right|\left|\mathbf{X}_{T'} - \mathbf{X}_E\right|\cos\phi \tag{50}$$

Substitute Eq. 49 and Eq. 50 into Eq. yields

$$\dot{\mathbf{V}} = \left|\mathbf{E}\right|\left|\dot{\mathbf{X}}_T\right|\cos\varphi - \left|\mathbf{E}\right|\lambda\cos\phi \tag{51}$$

Notes that the term $\left|\dot{\mathbf{X}}_T\right|\cos\varphi$ is the modulus speed projection of the target along the vector $\mathbf{E} = \mathbf{X}_T - \mathbf{X}_E$, and the term $\lambda\cos\phi$ is the modulus speed projection of the end-effector along the same vector, respectively.

The assumption that the speed of the end-effector is always greater than the target can be interpreted as

$$\left|\dot{\mathbf{X}}_T\right| < \lambda \tag{52}$$

From Eq. 44, Eq. 45 and Eq. 52 we obtain

$$\left|\dot{\mathbf{X}}_T\right|\cos\varphi < \left|\dot{\mathbf{X}}_T\right|\cos\phi < \lambda\cos\phi \tag{53}$$

Thus,

$$\dot{\mathbf{V}} = \left|\mathbf{E}\right|\left|\dot{\mathbf{X}}_T\right|\cos\varphi - \left|\mathbf{E}\right|\lambda\cos\phi \leq 0 \tag{54}$$

Therefore, the predictive kinematic control scheme is asymptotically stable.

Chapter 2
Design, Analysis, and Applications of Mobile Manipulators

Tao Song
Shanghai University, China

Feng Feng Xi
Ryerson University, Canada

Shuai Guo
Shanghai University, China

ABSTRACT

Presented in this chapter is a method for design and analysis of a mobile manipulator. The wrench induced by the movement of the robot arm will cause system tip-over or slip. In tip-over analysis, three cases are considered. The first case deals with the effect of the link weights and tip payload on the horizontal position of the CG. The second case deals with the effect of the joint speeds through the coupling terms including centrifugal forces and gyroscopic moments. The third case deals with the effect of the joint accelerations through the inertia forces and moments. In slip analysis, the first case considers the reaction force in relation to the stand-off distance between system and work-piece. The second and third cases investigate the effects of the joint speeds and accelerations. Then, the mobile platform is optimized to have maximum tip-over stability which optimizes the placement of the robot arm and accessory on the mobile platform. The effectiveness of the proposed method is demonstrated.

DOI: 10.4018/978-1-5225-5276-5.ch002

INTRODUCTION

A mobile manipulator integrates a mobile platform with an onboard robot arm called manipulator. Such a combined system is capable of executing manipulation tasks in a much larger workspace than that of a fixed-base manipulator. This combination offers the dexterous manipulation capability from the manipulator as well as the mobility from the mobile platform (Liu & Liu, 2010).

Mobile platforms can be classified as: tracked, legged, and wheeled. A tracked mobile manipulator has improved traction than other two due to a larger contact area with ground, making it ideal for working under unstructured environment for searching, rescuing, explosive ordnance disposing, mining, logging, farming, earth moving and planetary exploring (Liu & Liu, 2009). A legged mobile manipulator has the ability to step over obstacles along a moving path, uniquely suited for overcoming complex indoor environments including furniture, walls, stairs, doors, as well as outdoor environments that may be of rough terrain and uneven surfaces (Rehman, Focchi, & Lee, 2016). A wheeled platform, on the other hand, can move at a high speed on an even and solid surface in structured environments such as factory, store or home (Vysin & Knoflicek, 2003). This chapter deals with the wheeled manipulator used for manufacturing, such as drilling, riveting or line drawing.

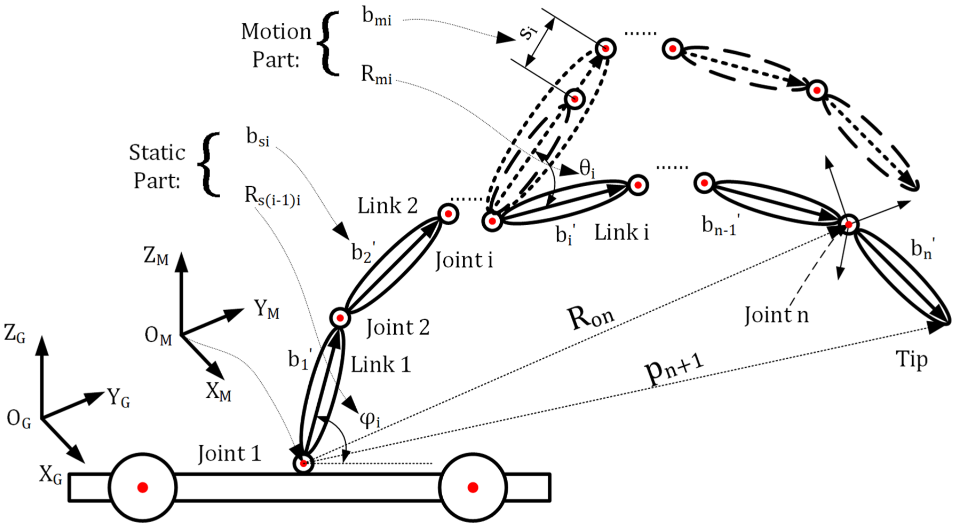
For a wheeled mobile manipulator, the wrench exerted by the manipulator onto the mobile platform will affect the system performance. These effects include tip-over or/and slip which are not considered for the fixed-base counterpart. Tip-over issue is of great concern for the safety of robot operation. The occurrence of slippage will lead to the loss of posture accuracy, thereby affecting manufacturing quality. The avoidance of these two issues should be studied in two aspects. The first one is to study the effect caused by the dynamics parameters of the manipulator to gain insights on the interaction with the mobile platform. Then the second one is to optimize the mobile manipulator system in order to obtain the maximum stability margin. In what follows, the details are provided.

WRENCH MODELING

Manipulator Kinematics

Figure 1 shows a kinematic model of the manipulator under this study. The method presented in (Xi, 2009; Lin, Xi, Mohamed, & Tu, 2010) is used here for kinematic modeling. This method formulates the manipulator kinematics through two parts. The first is a static part to represent the initial configuration of each link and the second is a motion part to represent the movement of each joint. For the static part,

Figure 1. Manipulator kinematic modeling with static and motion parts



a set of initial configuration set-up (ICSU) are defined including a static rotation matrix $\mathbf{R}_{si}$ and a static body vector $\mathbf{b}'_{si}$, for each link, that is

$$\mathbf{R}_{si} = \mathbf{R}_x\mathbf{R}_y\mathbf{R}_z \tag{1}$$

$$\mathbf{b}'_{si} = \mathbf{x}_i\mathbf{X}'_i + \mathbf{y}_i\mathbf{Y}'_i + \mathbf{z}_i\mathbf{Z}'_i \tag{2}$$

where $\mathbf{R}_x$, $\mathbf{R}_y$, $\mathbf{R}_z$ are three rotation matrices about X, Y, and Z axis of local frame i relative to local frame $i-1$, and $\mathbf{X}'_i$, $\mathbf{Y}'_i$, $\mathbf{Z}'_i$ are the three unit vectors of local frame i attached to the i th joint. $\mathbf{b}'_{si}$ is a vector representing the i th link at the initial configuration. In this chapter, a bold vector is expressed with respect to the base frame, and a bold vector with an apostrophe is expressed with respect to a local frame.

The total translation $\mathbf{b}'_i$ and total rotation $\mathbf{R}_{(i-1)i}$ of the i th link is expressed by including the motion part as

$$\mathbf{R}_{(i-1)i} = \mathbf{R}_{si}\mathbf{R}_{mi} \tag{3}$$

$$\mathbf{b}_i^{'} = \mathbf{b}_{si}^{'} + \mathbf{b}_{mi}^{'} \tag{4}$$

where $\mathbf{R}_{mi}$ and $\mathbf{b}_{mi}^{'}$ are the motional rotation matrix and motional body vector, respectively. In practice, $\mathbf{R}_{mi}$ corresponds to a rotational joint driven by a rotary motor and $\mathbf{b}_{mi}^{'}$ corresponds to a prismatic joint driven by a linear motor. In case of revolution joint, $\mathbf{R}_{mi}$ is equal to $\mathbf{R}_z\left(\theta_i\right)$, and θ_i is the rotation angle of the i th joint.

To this end, the end-effector's position $\mathbf{p}_{n+1}$ and orientation $\mathbf{R}_{on}$, can be expressed with respect to the base frame of the mobile platform as

$$\mathbf{p}_{n+1} = \sum_{i=0}^{n} \mathbf{R}_{oi} \mathbf{b}_i^{'} \tag{5}$$

$$\mathbf{R}_{on} = \prod_{i=1}^{n} \mathbf{R}_{(i-1)i} \tag{6}$$

where $\mathbf{R}_{oi}$ represents the rotation matrix of link i with respect to the base frame, which is multiplied sequentially by a number of $\mathbf{R}_{(i-1)i}$. In eqn. (5), subscript $n+1$ indicates the tip of the last link where the end-effector is located, and the end-effector's orientation coincides with the last link, expressed by $\mathbf{R}_{on}$.

Taking the time derivative of eqn. (5) and (6) leads to the following forward recursive velocity equations

$$\mathbf{v}_i = \mathbf{v}_{i-1} + \omega_{i-1} \times \mathbf{b}_{i-1} + \mathbf{R}_{oi-1} \dot{\mathbf{b}}_{i-1}^{'} \tag{7}$$

$$\omega_i = \omega_{i-1} + \omega_{i-1i} \tag{8}$$

where $\mathbf{v}_i$ is the velocity vector at the i th joint and ω_i is the angular velocity vector of the i th link. Note that the following holds

$$\dot{\mathbf{R}}_{i-1i} = \mathbf{R}_{si-1i} \dot{\mathbf{R}}_{mi} \tag{9}$$

$$\dot{\mathbf{b}}'_{\mathrm{i}} = \dot{\mathbf{b}}'_{\mathrm{mi}} \tag{10}$$

Furthermore, taking the time derivative of eqn. (7) and (8) results in the following forward recursive acceleration equations

$$\mathbf{a}_{\mathrm{i}} = \mathbf{a}_{\mathrm{i-1}} + \alpha_{\mathrm{i-1}} \times \mathbf{b}_{\mathrm{i-1}} + \omega_{\mathrm{i-1}} \times \left(\omega_{\mathrm{i-1}} \times \mathbf{b}_{\mathrm{i-1}}\right) + 2\left(\omega_{\mathrm{i-1}} \times \mathbf{R}_{\mathrm{oi-1}}\dot{\mathbf{b}}'_{\mathrm{i-1}}\right)\mathbf{R}_{\mathrm{oi-1}}\dot{\mathbf{b}}'_{\mathrm{i-1}} \tag{11}$$

$$\alpha_{\mathrm{i}} = \alpha_{\mathrm{i-1}} + \alpha_{\mathrm{i-1i}} \tag{12}$$

where $\mathbf{a}_{\mathrm{i}}$ is the acceleration vector at the i th joint and α_{i} is the angular acceleration vector of the i th link.

Manipulator Dynamics

A Newton-Euler formulation (Xi, 2009; Bianco, 2009) is applied to model the reaction forces and moments of the manipulator onto the mobile platform. Based on this formulation, the wrench acting on the i th joint can be expressed as

$$\mathbf{w}_{\mathrm{i}} = \mathbf{M}_{\mathrm{i}}\dot{\mathbf{t}}_{\mathrm{i}} + \mathbf{B}_{\mathrm{i}} + \mathbf{H}_{\mathrm{ii+1}}\mathbf{w}_{\mathrm{i+1}} \tag{13}$$

In eqn. (13), $\mathbf{w}_{\mathrm{i}}$ represents a wrench including a force vector $\mathbf{f}_{\mathrm{i}}$ and a moment vector $\mathbf{m}_{\mathrm{i}}$ on the i th joint and it is given below

$$\mathbf{w}_{\mathrm{i}} = \begin{bmatrix} \mathbf{f}_{\mathrm{i}} \\ \mathbf{m}_{\mathrm{i}} \end{bmatrix} = \left[\mathrm{f}_{\mathrm{iX}},\ \mathrm{f}_{\mathrm{iY}},\ \mathrm{f}_{\mathrm{iZ}},\ \mathrm{m}_{\mathrm{iX}},\ \mathrm{m}_{\mathrm{iY}},\ \mathrm{m}_{\mathrm{iZ}}\right]^{\mathrm{T}} \tag{14}$$

Generalized mass matrix $\mathbf{M}_{\mathrm{i}}$ is defined with respect to the i th joint as

$$\mathbf{M}_{\mathrm{i}} = \begin{bmatrix} \mathrm{m}_{\mathrm{i}}\mathbf{1} & \mathrm{m}_{\mathrm{i}}\tilde{\mathbf{b}}_{\mathrm{ic}}^{\mathrm{T}} \\ \mathrm{m}_{\mathrm{i}}\tilde{\mathbf{b}}_{\mathrm{ic}} & \mathbf{I}_{\mathrm{i}} \end{bmatrix} \tag{15}$$

where is the mass of link i, $\mathbf{b}_{\mathrm{ic}}$ is a vector representing center of gravity (CG) of the i th link relative to the i th joint; $\tilde{\mathbf{b}}_{\mathrm{ic}}$ is a skew matrix defined for $\mathbf{b}_{\mathrm{ic}}$, $\mathbf{1}$ is a

3x3 identity matrix, and $\mathbf{I}_{\mathrm{i}}$ is the matrix of inertia of moments of link i about joint i.

$\dot{\mathbf{t}}_{\mathrm{i}}$ is the vector including linear and angular acceleration vectors of the i th joint, given as

$$\dot{\mathbf{t}}_{\mathrm{i}} = \begin{bmatrix} \mathbf{a}_{\mathrm{i}} - \mathbf{g} \\ \boldsymbol{\alpha}_{\mathrm{i}} \end{bmatrix} \tag{16}$$

where $\mathbf{a}_{\mathrm{i}}$ and $\boldsymbol{\alpha}_{\mathrm{i}}$ are defined in eqns. (11) and (12), and $\mathbf{g}$ is the gravitational acceleration vector.

$\mathbf{B}_{\mathrm{i}}$ is a matrix representing the coupling terms including centrifugal forces as well as gyroscopic moments given as

$$\mathbf{B}_{\mathrm{i}} = \begin{bmatrix} \mathrm{m}_{\mathrm{i}}\omega_{\mathrm{i}} \times \left(\omega_{\mathrm{i}} \times \mathbf{b}_{\mathrm{ic}}\right) \\ \omega_{\mathrm{i}} \times \left(\mathbf{I}_{\mathrm{i}}\omega_{\mathrm{i}}\right) \end{bmatrix} \tag{17}$$

where ω_{i} is defined in eqn. (8). Note that the Coriolis term $2\left(\omega_{\mathrm{i-1}} \times \mathbf{R}_{\mathrm{oi-1}}\dot{\mathbf{b}}'_{\mathrm{i-1}}\right)$ in this formulation is included in joint acceleration, as indicated in (11).

Finally $\mathbf{H}_{\mathrm{ii+1}}$ represents a wrench transformation matrix from the $\left(\mathrm{i}+1\right)$ th joint to the i th joint as

$$\mathbf{H}_{\mathrm{ii+1}} = \begin{bmatrix} 1 & 0 \\ \tilde{\mathbf{b}}_{\mathrm{i}} & 1 \end{bmatrix} \tag{18}$$

where $\tilde{\mathbf{b}}_{\mathrm{i}}$ is a skew matrix defined for $\mathbf{b}_{\mathrm{i}}$.

Eqn. (13) provides a backward recursion for computing the forces and moments from the tip numbered by $\mathrm{n}+1$ to the base membered by 1. First, for $\mathrm{i}=\mathrm{n}$, the wrench of the last joint, $\mathbf{w}_{\mathrm{n}}$, can be computed from the wrench acting on the tip $\mathbf{w}_{\mathrm{n+1}}$, given as

$$\mathbf{w}_{\mathrm{n+1}} = \begin{bmatrix} \mathbf{f}_{\mathrm{Tip}} \\ \mathbf{m}_{\mathrm{Tip}} \end{bmatrix} = \left[\mathrm{f}_{\mathrm{TipX}},\ \mathrm{f}_{\mathrm{TipY}},\ \mathrm{f}_{\mathrm{TipZ}},\ \mathrm{m}_{\mathrm{TipX}},\ \mathrm{m}_{\mathrm{TipY}},\ \mathrm{m}_{\mathrm{TipZ}}\right]^{\mathrm{T}} \tag{19}$$

This recursive computation goes through all the intermediate joints till $\mathrm{i}=1$ to obtain $\mathbf{w}_1$. The reaction wrench (forces and moments) from the manipulator to the mobile platform is expressed as $-\mathbf{w}_1$.

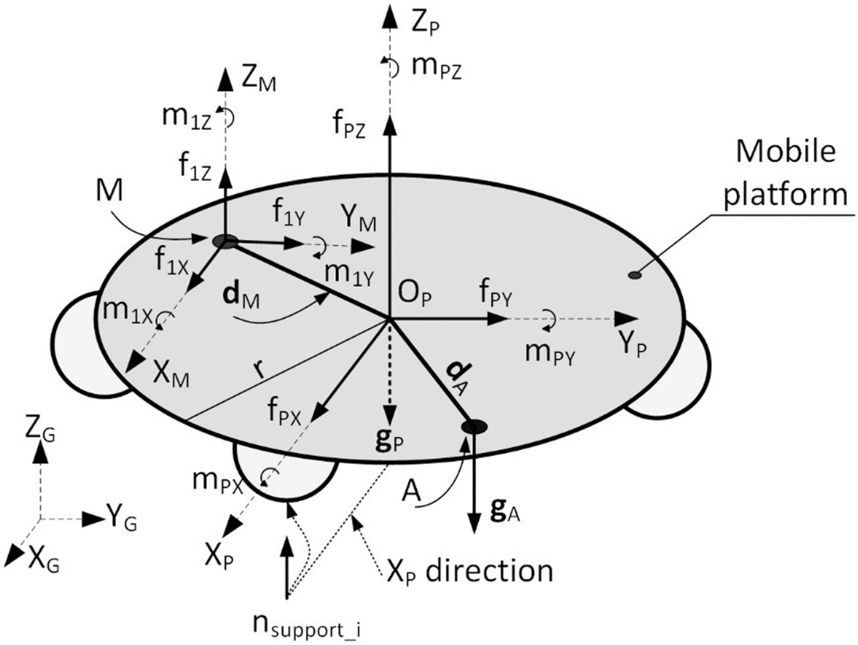
There are six components in this wrench, i.e. three force components and three moment components. The three out-of-plane wrench components (f_z, m_x, m_y) make the system tip-over, and the three in-plane wrench components (f_x, f_y, m_z) make system slip. The tip-over and slip issues will be discussed in following sections.

TIP-OVER ANALYSIS

Tip-Over Criterion

Figure 2 depicts the underlying problem. In this figure, the ellipse represents a mobile platform. A local coordinate frame $\left\{\mathrm{X_P}, \mathrm{Y_P}, \mathrm{Z_P}\right\}$ is attached to the center of gravity of the platform at point $\mathrm{O_P}$. $\mathbf{g}_\mathrm{P}$ is the vector of the gravitational force of

Figure 2. Wrench on mobile platform



the mobile platform and $\mathrm{m_P}$ is its mass. The mass of the selected manipulator is $\mathrm{m_r}$. It is mounted at point M to which the manipulator's base coordinate frame $\left\{\mathrm{X_M}, \mathrm{Y_M}, \mathrm{Z_M}\right\}$ is attached. The reaction wrench, denoted by $-\mathbf{w}_1$, is acting on point M from the manipulator onto the mobile platform. This wrench has three force components $\mathbf{f}_\mathrm{M} = \left[\mathrm{f_{1X}},\ \mathrm{f_{1Y}},\ \mathrm{f_{1Z}}\right]^\mathrm{T}$ and three moment components $\mathbf{m}_\mathrm{M} = \left[\mathrm{m_{1X}},\ \mathrm{m_{1Y}},\ \mathrm{m_{1Z}}\right]^\mathrm{T}$, i.e. $-\mathbf{w}_1 = \left[\mathbf{f}_\mathrm{M}{}^\mathrm{T},\ \mathbf{m}_\mathrm{M}{}^\mathrm{T}\right]^\mathrm{T}$.

When tip-over occurs, the mobile platform rolls about a tip-over axis formed by two adjacent wheels. Two vectors ${}^\mathrm{n}\mathbf{p}_\mathrm{i}$ and ${}^\mathrm{n}\mathbf{p}_\mathrm{i+1}$ represent the positions of two adjacent wheels of the mobile platform, forming a tip-over axis $\mathbf{e}_\mathrm{ii+1}$. $\mathbf{d}_\mathrm{M}$ is the vector from the origin $\mathrm{O_P}$ to point M, representing the mounting position of the manipulator. $\mathbf{d}_\mathrm{ii+1}$ is a vector from point M to the tip-over axis $\mathbf{e}_\mathrm{ii+1}$, while $\mathbf{l}_\mathrm{ii+1}$ is a vector from $\mathrm{O_P}$ to the tip-over axis $\mathbf{e}_\mathrm{ii+1}$. The tip-over moment (TOM) about the tip-over axis $\mathbf{e}_\mathrm{ii+1}$ (Guo, Song, Xi, & Mohamed, 2017) can be derived considering all the afore-mentioned forces and moments as

$$\mathrm{TOM_{ii+1}} = \mathbf{m}_\mathrm{M} \cdot \mathbf{e}_\mathrm{ii+1} + \left(\mathbf{f}_\mathrm{M} \times \mathbf{d}_\mathrm{ii+1}\right) \cdot \mathbf{e}_\mathrm{ii+1} + \left(\mathbf{g}_\mathrm{P} \times \mathbf{l}_\mathrm{ii+1}\right) \cdot \mathbf{e}_\mathrm{ii+1} \tag{20}$$

Generally for a n wheel mobile manipulator with supporting wheels forming convex closed hull, there are n tip-over axes, i.e. nTOM values. The maximum $\mathrm{TOM_{ii+1}}$ value represents that around axis $\mathbf{e}_\mathrm{ii+1}$ the system experiences the biggest tip-over moment. If this maximum TOM value is positive, the system is tipping over, and the tip over moment is TOM. If this maximum TOM value is negative, the system is stable with a tip-over moment margin equal to $\left|\mathrm{TOM}\right|$.

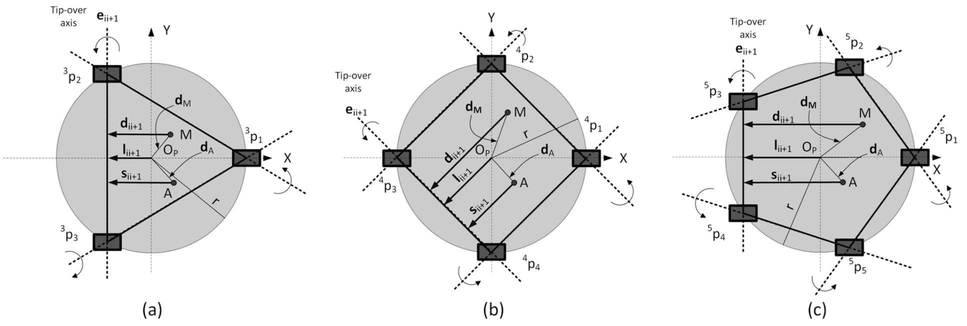
In detail, for a three to five wheeled mobile manipulator which are common used, the corresponding parameters in eqn. (20) can be obtained from Figure 3 (a) to (c).

For convenience, the X axis is defined along with the line between the origin $\mathrm{O_P}$ and the first wheel. The coordinates of each support point of the evenly distributed wheels can be expressed as

$${}^\mathrm{n}\mathbf{p}_\mathrm{i} = \left[\mathrm{r} \cdot \cos\left(\frac{\mathrm{i}-1}{\mathrm{n}} \cdot 2\pi\right),\ \mathrm{r} \cdot \sin\left(\frac{\mathrm{i}-1}{\mathrm{n}} \cdot 2\pi\right), 0\right]^\mathrm{T} \qquad \mathrm{i} = \left\{1, 2, \ldots, \mathrm{n}\right\} \tag{21}$$

where ${}^\mathrm{n}\mathbf{p}_\mathrm{i}$ represents the position of the i th wheel for a n-wheel platform.

Figure 3. The different distributions of the support wheels



All the vectors in Figure 3 are defined as follows. The tip-over axis denoted by $\mathbf{e}_{ii+1}$ is defined by the vectors of the two adjacent support points (i.e. wheels) as

$$\mathbf{e}_{ii+1} = \left({}^{n}\mathbf{p}_{i+1} - {}^{n}\mathbf{p}_{i}\right) / \left|{}^{n}\mathbf{p}_{i+1} - {}^{n}\mathbf{p}_{i}\right| \tag{22}$$

In this study, when index number i is equal to n, $i+1$ is rounded back to 1. Next, the vector $\mathbf{l}_{ii+1}$ from origin O_P to the tip-over axis $\mathbf{e}_{ii+1}$ is defined as

$$\mathbf{l}_{ii+1} = {}^{n}\mathbf{p}_{i} - \left({}^{n}\mathbf{p}_{i} \cdot \mathbf{e}_{ii+1}\right)\mathbf{e}_{ii+1} \tag{23}$$

As for the placement of the manipulator, $\mathbf{d}_{M}$ is the vector of point M from origin O_P, i.e. the mounting position of the manipulator. The vector $\mathbf{d}_{ii+1}$ from M point to tip-over axis $\mathbf{e}_{ii+1}$ is defined as

$$\mathbf{d}_{ii+1} = \left({}^{4}\mathbf{p}_{i} - \mathbf{d}_{M}\right) - \left(\left({}^{4}\mathbf{p}_{i} - \mathbf{d}_{M}\right) \cdot \mathbf{e}_{ii+1}\right)\mathbf{e}_{ii+1} \tag{24}$$

With these equations defined, all $TOMs$ about all tip-over axes can be obtained. The sign of TOM can show whether the system is stable. The positive value represents the system is tipping-over with the moment TOM. The negative value represents the system is stable and the tip-over moment margin of TOM. The axis around which the system does or will tip-over can be obtained in the course of computation.

Static Case

For static case, i.e. $\dot{\theta}_i = 0$, and $\ddot{\theta}_i = 0$, the purpose is to investigate the effect of gravitational forces. In this case, only the payload on the manipulator's end-effector and the weights of the manipulator's links are considered. Referring to the recursive wrench eqn. (13), only the first term and the last term remain. Hence, this equation is reduced to

$$^{s}\mathbf{w}_{i} = {}^{s}\mathbf{w}_{gi} + \mathbf{w}_{pi} i = \{1,2,3\} \tag{25}$$

where left superscript s indicates the static case and the other terms are expressed as

$$^{s}\mathbf{w}_{gi} = \mathbf{M}_{i}\, {}^{s}\dot{\mathbf{t}}_{i} \tag{26}$$

$$\mathbf{w}_{pi} = \mathbf{H}_{ii+1}\, {}^{s}\mathbf{w}_{i+1} \tag{27}$$

$$^{s}\dot{\mathbf{t}}_{i} = \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix} \tag{28}$$

Using eqn. (26) for $\mathbf{M}_i$ and multiplying it by $^{s}\dot{\mathbf{t}}_{i}$, it can be shown that term $^{s}\mathbf{w}_{gi}$ is defined by a gravitational force of the weight of link i, and its moment from the CG of link i to joint i. The term $\mathbf{w}_{pi}$ results from upper link $i+1$. The payload is expressed by $m_{payload}$ and included in wrench $^{s}\mathbf{w}_{n+1}$ as

$$^{s}\mathbf{w}_{n+1} = [0,\, 0,\, -m_{payload} g,\, 0,\, 0,\, 0] \tag{29}$$

The reaction wrench $^{s}\mathbf{w}_{1}$ can be obtained by using the recursive eqn. (25)

$$^{s}\mathbf{w}_{1} = {}^{s}\mathbf{w}_{g1} + \mathbf{w}_{p1} \tag{30}$$

where

$$
{}^{s}\mathbf{w}_{g1} = \mathbf{M}_{1}\,{}^{s}\dot{\mathbf{t}}_{1} + \mathbf{H}_{12}\mathbf{M}_{2}\,{}^{s}\dot{\mathbf{t}}_{2} + \mathbf{H}_{12}\mathbf{H}_{23}\mathbf{M}_{3}\,{}^{s}\dot{\mathbf{t}}_{3} + \ldots + \left(\prod_{i=1}^{n}\mathbf{H}_{i-1i}\right)\mathbf{M}_{n}\,{}^{s}\dot{\mathbf{t}}_{n} \tag{31}
$$

$$
\mathbf{w}_{p1} = \left(\prod_{i=1}^{n+1}\mathbf{H}_{i-1i}\right)\mathbf{w}_{p(n+1)} \tag{32}
$$

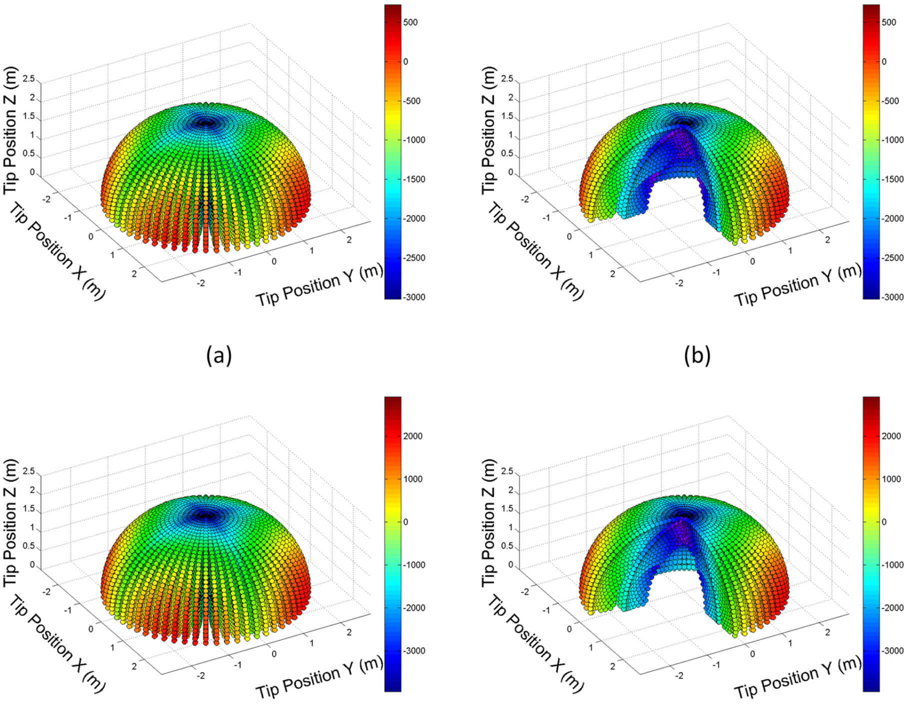
Eqn. (31) represents the reaction forces and moments ${}^{s}\mathbf{w}_{g1}$ caused by the weights of the manipulator's links. For the manipulator under study, this wrench results from the gravitational forces of three links, $m_3 g$, $m_2 g$, $m_1 g$, all pointing downwards. For three out-of-plane components associated with stability, force f_{1Z} is the summation of the three gravitational forces, while moments m_{1X} and m_{1Y}, are the products of the three gravitational forces and their CG distances in Y and X direction, respectively.

Eqn. (32) represents the reaction forces and moments caused by the payload. The gravitational force due to the payload is $m_{payload} g$, pointing downwards. For the three out-of-plane components associated with stability; force f_{1Z} is equal to this gravitational force, while moments m_{1X} and m_{1Y} are the products of the gravitational force and the distances in Y and X direction.

To sum up, since the gravitational forces of the links and the payload are always downwards, the three out-of-plane wrench components pertaining to the tip-over instability only correlate with the horizontal positions of link CGs and the manipulator's tip. Based on the modeling presented in this chapter, simulations are carried out over the manipulator workspace.

Figure 4 shows the simulation results based on eqn. (30) for the static case. Figure 4 (a) is for link weights only, i.e. no payload at the tip point. Figure 4 (c) is for link weights with payload at the tip point. Figure 4 (b) and (d) are cut-off views of Figure 4 (a) and (c), individually. Figure 4 (a) and (c) consider TOM of the system all over the workspace. The red color represents a big TOM, while the blue color represents a small TOM value. As mentioned before the positive TOM value represents that the system is unstable while negative TOM value represents that the system is stable with a TOM margin at that position.

In non-payload case, when the tip is at the furthest position from the center of the platform in X-Y plane, the maximum TOM becomes positive. It can be seen that when the tip of manipulator is far away from the center of the platform, there is more mass of manipulator close to the boundary of the platform, so TOM value is big. In contrast, when the tip of manipulator is close to the center of the platform, *Figure 4. (a) TOM results over workspace for static case without payload, (b) Cut-off view of (a) (c) TOM results over workspace for static case with payload, (d) Cut-off view of (c)*



(a)

(b)

(c)

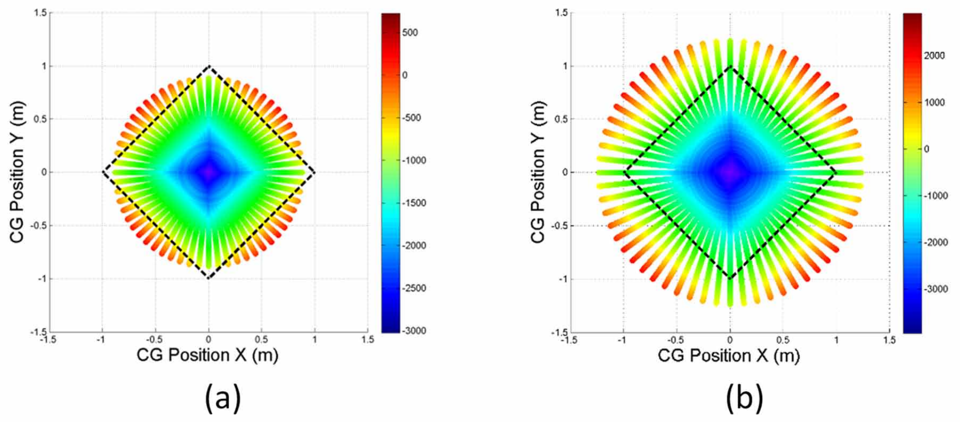
(d)

the most mass of manipulator is close to the center, so the system is stable with a TOM margin. The minimum TOM of the whole workspace becomes negative when X and Y coordinates of system's CG are near the origin point.

In payload case, when the tip is at the farthest position as the non-payload case the maximum positive TOM becomes bigger. In this case, the stability with payload is worse than the one without payload. However, when X and Y coordinates of system's CG including payload are near the origin point, the minimum is negative. At this point, the payload makes the system more stable just like increasing the mass of the platform.

As mentioned before, the stability problem in the static case is only affected by the horizontal positions of the link CGs and the payload. TOM in terms of system CG is simulated with the result shown in Figure 5. Figure 5 (a) is the result of non-

Figure 5. TOM for static case in terms of system CG



payload case, while Figure 5 (b) is the result of payload case. Under this plane view, the workspace becomes a circle. When there is a payload at the tip of manipulator, the system's CG changes wider than non-payload case. In Figure 5, the four black lines represent the tip-over axes formed by four wheels. The stability of the system depends on the CG position of system. It can be easily observed that TOM has a close correlation with the distance of CG from origin point. When the CG is out of the area formed by four tip-over axes, called bounding box, TOM is positive, system unstable. If within the bounding box, TOM is negative, system stable.

Joint Speed Case

For joint speed case, i.e. $\dot{\theta}_{\mathrm{i}} \neq 0$, and $\ddot{\theta}_{\mathrm{i}} = 0$, the purpose is to investigate the effect of the coupling term including centrifugal forces and gyroscopic moments on tip-over stability. With only joint speeds considered, eqn. (13) can be re-written as

$$^{\mathrm{v}}\mathbf{w}_{\mathrm{i}} = {}^{\mathrm{s}}\mathbf{w}_{\mathrm{gi}} + \mathbf{w}_{\mathrm{pi}} + {}^{\mathrm{v}}\mathbf{w}_{\mathrm{mi}} \tag{33}$$

where $^{\mathrm{s}}\mathbf{w}_{\mathrm{gi}}$ and $\mathbf{w}_{\mathrm{pi}}$ are already defined in eqn. (30), the velocity term $^{\mathrm{v}}\mathbf{w}_{\mathrm{mi}} = {}^{\mathrm{v}}\mathbf{w}_{\mathrm{Ci}} + {}^{\mathrm{v}}\mathbf{w}_{\mathrm{vi}}$ in which $^{\mathrm{v}}\mathbf{w}_{\mathrm{Ci}}$ and $^{\mathrm{v}}\mathbf{w}_{\mathrm{vi}}$ can be derived from eqn. (15) to eqn. (17) as

$$^{\mathrm{v}}\mathbf{w}_{\mathrm{Ci}} = \mathbf{M}_{\mathrm{i}}\, {}^{\mathrm{v}}\dot{\mathbf{t}}_{\mathrm{Ci}} \tag{34}$$

$$ {}^{\mathrm{v}}\mathbf{w}_{\mathrm{vi}} = \mathbf{B}_{\mathrm{i}} \tag{35} $$

Since the industrial robot under this study is only made of revolute joints, the joint acceleration vector in eqn. (11) for joint speed only case becomes

$$ \mathbf{a}_{\mathrm{i}} = \mathbf{a}_{\mathrm{i-1}} + \omega_{\mathrm{i-1}} \times \left(\omega_{\mathrm{i-1}} \times \mathbf{b}_{\mathrm{i-1}}\right) \tag{36} $$

and

$$ {}^{\mathrm{v}}\dot{\mathbf{t}}_{\mathrm{Ci}} = \begin{bmatrix} \mathbf{a}_{\mathrm{i}} \\ 0 \end{bmatrix} \tag{37} $$

hence,

$$ {}^{\mathrm{v}}\dot{\mathbf{t}}_{\mathrm{Ci}} = \begin{bmatrix} \sum_{1}^{\mathrm{i-1}} \omega_{\mathrm{k}} \times \left(\omega_{\mathrm{k}} \times \mathbf{b}_{\mathrm{k}}\right) \\ 0 \end{bmatrix} \tag{38} $$

Looking back at eqn. (34), ${}^{\mathrm{v}}\mathbf{w}_{\mathrm{Ci}}$ only has the top three components representing the sum of the centrifugal forces of link i relative joint 1 to $\mathrm{i}-1$. For eqn. (35), the top three components of ${}^{\mathrm{v}}\mathbf{w}_{\mathrm{vi}}$ represent the centrifugal force of link i itself relative to joint i, while the bottom three components of ${}^{\mathrm{v}}\mathbf{w}_{\mathrm{vi}}$ represent the gyroscopic moments on link i.

Through recursive manipulation, it can be shown that the reaction wrench on the platform in the joint speed case is determined as

$$ {}^{\mathrm{v}}\mathbf{w}_{1} = {}^{\mathrm{s}}\mathbf{w}_{1} + {}^{\mathrm{v}}\mathbf{w}_{\mathrm{C1}} + {}^{\mathrm{v}}\mathbf{w}_{\mathrm{v1}} \tag{39} $$

where ${}^{\mathrm{s}}\mathbf{w}_{1}$ is already derived in eqn. (30) for the static case, ${}^{\mathrm{v}}\mathbf{w}_{\mathrm{C1}}$ and ${}^{\mathrm{v}}\mathbf{w}_{\mathrm{v1}}$ due to joint speed are given as

$$ {}^{\mathrm{v}}\mathbf{w}_{\mathrm{C1}} = \mathbf{M}_{1}\,{}^{\mathrm{v}}\dot{\mathbf{t}}_{\mathrm{C1}} + \mathbf{H}_{12}\mathbf{M}_{2}\,{}^{\mathrm{v}}\dot{\mathbf{t}}_{\mathrm{C2}} + \mathbf{H}_{12}\mathbf{H}_{23}\mathbf{M}_{3}\,{}^{\mathrm{v}}\dot{\mathbf{t}}_{\mathrm{C3}} + \ldots + \left(\prod_{\mathrm{i=1}}^{\mathrm{n}} \mathbf{H}_{\mathrm{i-1i}}\right)\mathbf{M}_{\mathrm{n}}\,{}^{\mathrm{v}}\dot{\mathbf{t}}_{\mathrm{Cn}} \tag{40} $$

$$ {}^{v}\mathbf{w}_{v1} = \mathbf{B}_1 + \mathbf{H}_{12}\mathbf{B}_2 + \mathbf{H}_{12}\mathbf{H}_{23}\mathbf{B}_3 + \ldots + \left(\prod_{i=1}^{n}\mathbf{H}_{i-1i}\right)\mathbf{B}_n \tag{41} $$

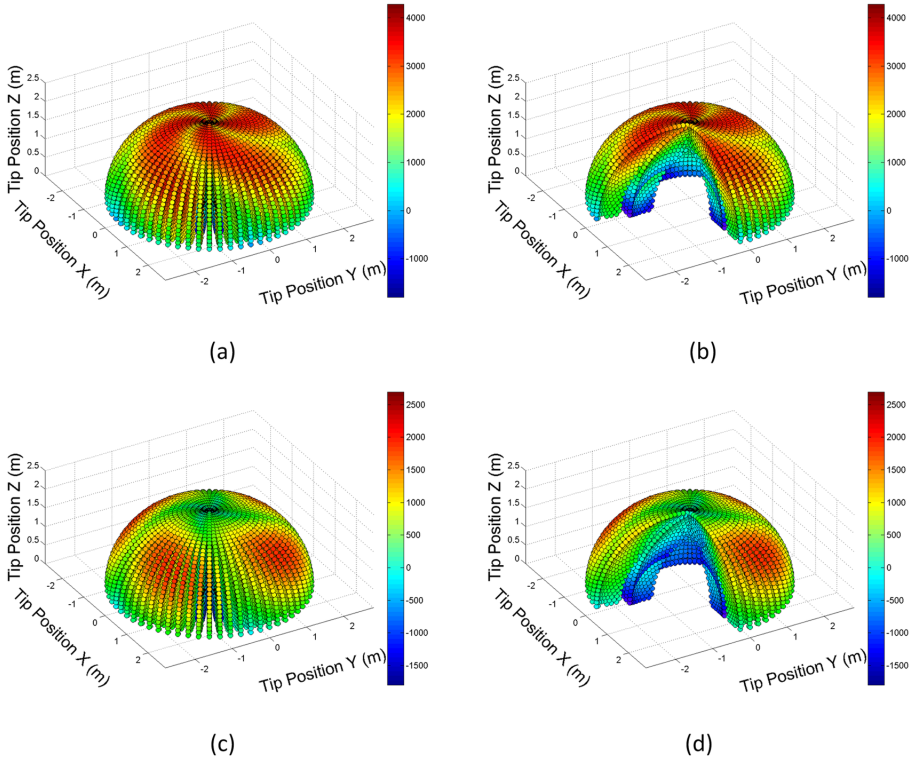
Eqns. (40) and (41) show that in addition to ${}^{s}\mathbf{w}_1$, ${}^{v}\mathbf{w}_1$ is obtained by accumulating the coupling term of each link through corresponding wrench transformations to the base. As explained before, term ${}^{v}\mathbf{w}_{C1}$ and ${}^{v}\mathbf{w}_{v1}$ contain centrifugal forces and gyroscopic moments. Unlike gravitational forces, centrifugal forces do not always point downwards. While the vertical parts of these forces will contribute to f_{1Z}, the horizontal parts will contribute to the two out-of-plane reaction moments, m_{1X} and m_{1Y}, therefore the tip-over stability will be affected by the vertical positions of link CGs. In addition, gyroscopic moments will also contribute to reaction moment m_{1X} and m_{1Y}.

Joint 1 rotates in the horizontal plane and the effect would be identical in both directions. For joint 2 and 3, difference cases will be either joints 2 and 3 in the same direction or joints 2 and 3 in opposite directions. Therefore, only the last two cases are considered for simulation.

Figure 6 (a) and (c) show TOM of the whole workspace, while Figure 6 (b) and (d) are the cut-off views of Figure 6 (a) and (c). Figure 6 (a) and (b) show the simulation result for joint speed = [175, 175, 175] $^{\circ}/\,\mathrm{s}$, i.e. joints 2 and 3 rotating in the same directions. The color and value of TOM is defined as before. In this case, the maximum TOM is positive, and the minimum TOM is negative. Figure 6 (c) and (d) show the simulation result for joint speed = [175, 175, - 175] $^{\circ}/\,\mathrm{s}$, i.e. joints 2 and 3 rotating in the opposite directions. In this case, the maximum TOM is positive but smaller than former one, and the minimum TOM is equal to the former one. Though the minimum TOM in two cases are equal, it can be seen from other two value and these figures that the latter has a smaller overall TOM than the former in the whole workspace. The reason for this difference is that the opposite joint rotations of joints 2 and 3 will reduce the absolute angular speed of link 3, hence reducing centrifugal force and gyroscopic moment.

As illustrated in Figure 7 (a), the first joint rotation will create centrifugal forces $\mathbf{f}_{1C}$, $\mathbf{f}_{2C}$, and $\mathbf{f}_{3C}$ on link 1, 2 and 3, all in horizontal direction, solely contributing to reaction moments m_{1X} and m_{1Y}, which increases TOM, thereby affecting tip-over stability. As shown in Figure 7 (b) and (c), the resulting centrifugal forces in these two cases do not include downstream links and they are not completely in the horizontal direction, hence the contribution to TOM is reduced, which would make the system more stable. Figure 7 (b) shows that the second joint rotation will create

Figure 6. TOM over workspace for joint speed case (a), (b) with the speeds of the joints [175, 175, 175] $^{\circ}/\mathrm{s}$ *(c), (d) with the speeds of the joints [175, 175, -175]* $^{\circ}/\mathrm{s}$



(a)

(b)

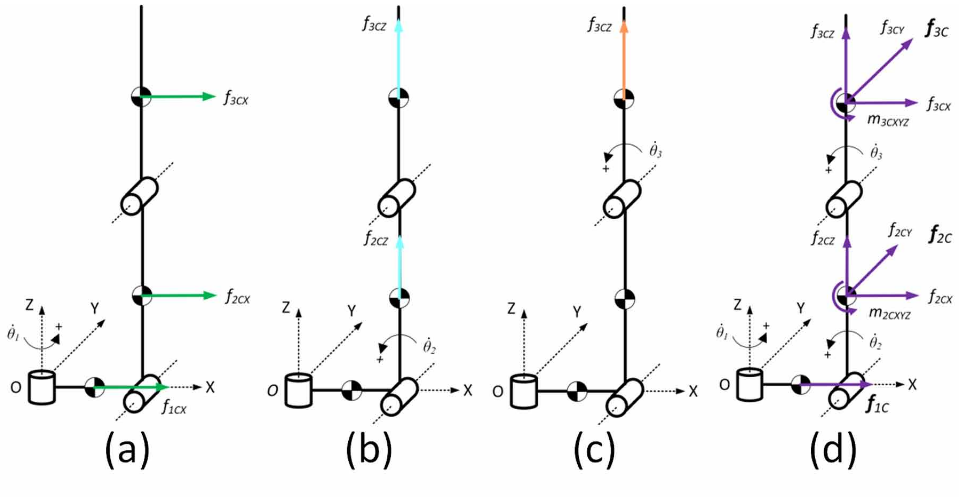
(c)

(d)

centrifugal forces, $\mathbf{f}_{2C}$ and $\mathbf{f}_{3C}$ on link 2 and 3, while Figure 7 (c) shows the force on link 3 only for joint 3 rotation.

As shown in Figure 7 (d), the first two joint rotations will create centrifugal force $\mathbf{f}_{2C}$ and gyroscopic moment $\mathbf{m}_{2C}$ on link 2. All the three joint rotations will create centrifugal force $\mathbf{f}_{3C}$ and gyroscopic moment $\mathbf{m}_{3C}$ on link 3, both determined from the absolute angular velocity, ω_3, which is composed of joint 2 and 3 speeds. Hence the centrifugal force and gyroscopic moment are bigger when the directions of joint 2 and 3 speeds are the same; smaller when they are opposite. Therefore, one observation that can be drawn from this simulation is that the rotation of two coplanar joints would worsen the tip-over stability if they rotate in the same directions.

Figure 7. Centrifugal forces and gyroscopic moments derived from joint speed case (a) joint 1 rotation, (b) joint 2 rotation, (c) joint 3 rotation, (d) all three joint rotations



Joint Acceleration Case

For joint acceleration case, i.e. $\dot{\theta}_i = 0$, and $\ddot{\theta}_i \neq 0$, the purpose is to investigate the effect of inertia forces and moments on tip-over stability. In this case assuming that all joint speeds are equal to zero, only joint accelerations are considered, equivalent to when the manipulator either starts or stops. A sudden start or stop would cause a tip-over problem. Under this assumption, eqn. (13) becomes

$$ {}^{a}\mathbf{w}_{i} = {}^{s}\mathbf{w}_{gi} + \mathbf{w}_{pi} + {}^{m}\mathbf{w}_{ai} \tag{42} $$

where ${}^{s}\mathbf{w}_{gi}$ and $\mathbf{w}_{pi}$ are already defined in eqn. (30), the acceleration term ${}^{m}\mathbf{w}_{ai}$, with its top three components representing inertia forces and bottom three representing inertia moments, is defined as below

$$ {}^{m}\mathbf{w}_{ai} = \mathbf{M}_{i}\,{}^{m}\dot{\mathbf{t}}_{i} \tag{43} $$

In eqn. (43), ${}^{m}\dot{\mathbf{t}}_{i}$ is the vector of linear and angular acceleration of the i th link given as

$$^{m}\dot{\mathbf{t}}_{i} = \begin{bmatrix} \mathbf{a}_{i} \\ \alpha_{i} \end{bmatrix} \tag{44}$$

where $\mathbf{a}_i$ and α_i are defined in eqns. (11) and (12), respectively.

Because of no joint speed, term $\mathbf{B}_i$ is zero. The wrench transformation $\mathbf{H}_{ii+1}\mathbf{w}_{i+1}$ is the same as before. To this end, it can be shown that the reaction wrench in this case, $^{a}\mathbf{w}_{1}$, can be determined as

$$^{a}\mathbf{w}_{1} = {}^{s}\mathbf{w}_{1} + {}^{m}\mathbf{w}_{a1} \tag{45}$$

where $^{s}\mathbf{w}_{1}$ is defined before and $^{m}\mathbf{w}_{a1}$ due to joint accelerations is given as

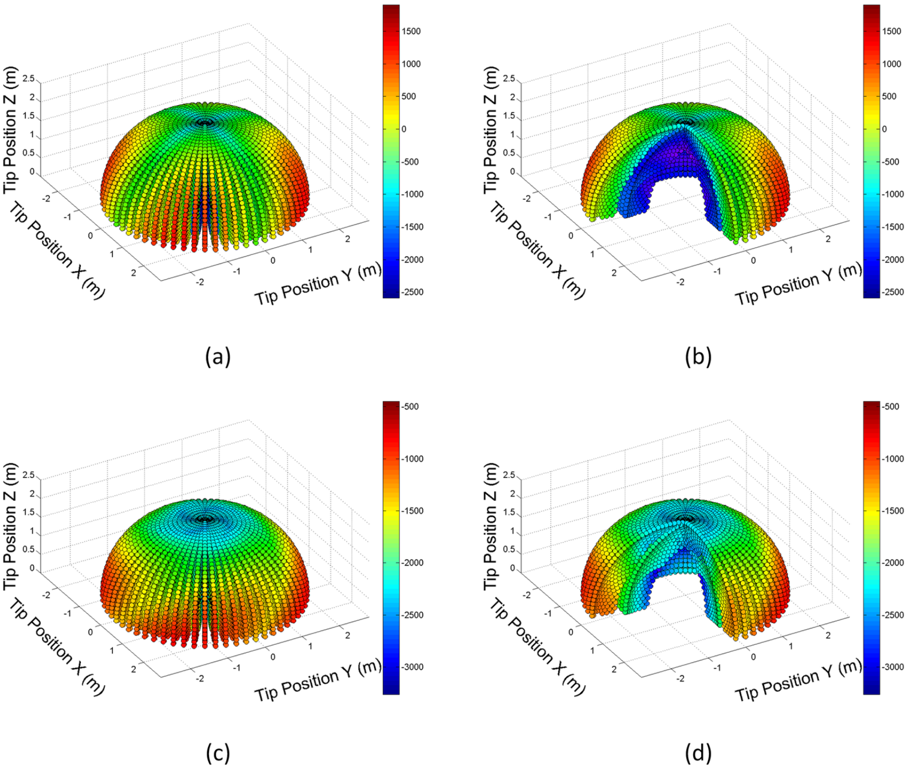
$$^{m}\mathbf{w}_{a1} = \mathbf{M}_{1}\,{}^{m}\dot{\mathbf{t}}_{1} + \mathbf{H}_{12}\mathbf{M}_{2}\,{}^{m}\dot{\mathbf{t}}_{2} + \mathbf{H}_{12}\mathbf{H}_{23}\mathbf{M}_{3}\,{}^{m}\dot{\mathbf{t}}_{3} + \ldots + \left(\prod_{i=1}^{n}\mathbf{H}_{i-1i}\right)\mathbf{M}_{n}\,{}^{m}\dot{\mathbf{t}}_{n} \tag{46}$$

Eqn. (46) reveals that $^{m}\mathbf{w}_{a1}$ is obtained by accumulating the inertia term of each link through corresponding wrench transformations to the base. Physically, term $^{m}\mathbf{w}_{a1}$ represents inertia forces and inertial moments. Similar to centrifugal forces, inertia forces do not always point downwards. While the vertical parts of these forces will contribute to f_{1Z}, the horizontal parts will contribute to the two out-of-plane reaction moments, m_{1X} and m_{1Y}. Therefore, the tip-over stability will also be affected by the vertical positions of link CGs. Furthermore, inertia moments will also contribute to the reaction moments m_{1X} and m_{1Y}.

To simply this, only the maximum joint accelerations are considered in two rotation directions, positive and negative in this joint acceleration case.

Figure 8 (a) and (c) show TOM of the whole workspace, while Figure 8 (b) and (d) are the cut-off views of Figure 8 (a) and (c). Figure 8 (a) and (b) show the simulation results for joint accelerations = [200, 200, 200] $^{\circ}/\mathrm{s}^{2}$, i.e. joint 2 and 3 both rotating in the positive direction, while Figure 8 (c) and (d) show the results for joint accelerations = [200, -200, -200] $^{\circ}/\mathrm{s}^{2}$, i.e. joint 2 and 3 in the negative direction. The color and value of TOM is defined as before. It can be seen from these figures that the latter has a smaller maximum TOM value than the former one, which is the same as the mean TOM value. The reason for this is that the directions of joint 2 and 3 acceleration affect tip-over. The positive joint accelerations of joint

Figure 8. TOM over workspace for joint acceleration case (a) joint accelerations [200, 200, 200] $^\circ/\mathrm{s}^2$*, (b) Cut-off view of the (a) (c) joint accelerations [200, -200, -200]* $^\circ/\mathrm{s}^2$*, (d) Cut-off view of the (c)*



(a)

(b)

(c)

(d)

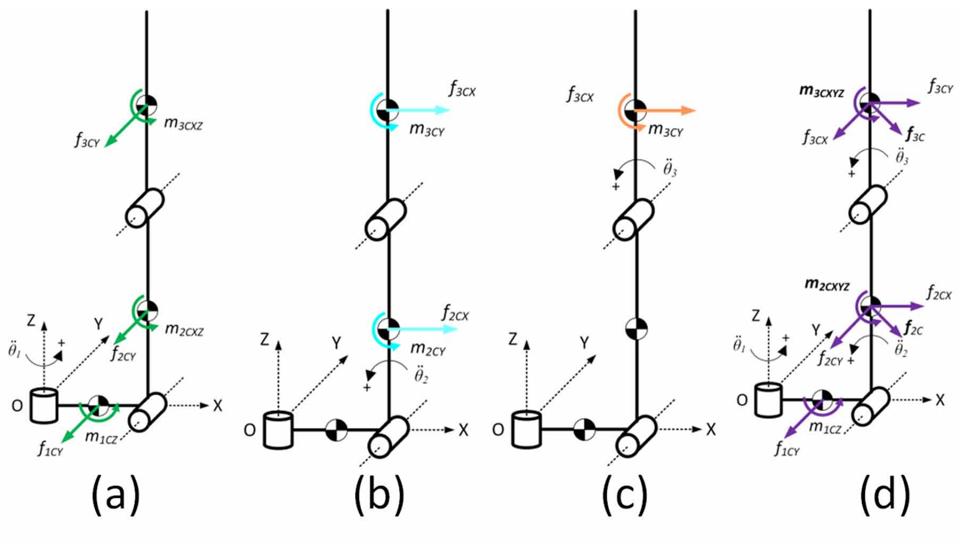
2 and 3 would increase the value of the tip-over stability index TOM just as what gravitational acceleration does, while the negative joint accelerations would reduce this value just like gravitational acceleration.

As shown in Figure 9 (a), the first joint acceleration in either positive or negative direction will create inertia forces f_{1CY}, f_{2CY}, f_{3CY}, and inertia moment m_{1CZ}, m_{2CXZ}, m_{3CXZ} on each of three links, respectively. Furthermore, as shown in Figure 11, consider two positions of link 1, i.e. position 1 and 2 symmetric along X axis, both under joint 1 acceleration; then respective inertial force $\mathbf{f}_{1C1}$ and $\mathbf{f}_{1C2}$ would have the same component along Y axis but in opposite directions along X axis, which will result in different TOM, causing asymmetry. This asymmetry phenomenon is caused by accelerations in the horizontal plane. Comparing the TOM values for only joint 1 acceleration to the TOM values in statics case, the joint 1 acceleration contributes little on TOM value. The reason for this is that inertia moments m_{1CZ}, m_{2CXZ}, m_{3CXZ} caused by joint 1 acceleration which is in X-Y plane do not contribute to TOM, i.e. not making the system unstable.

Figure 9 (b) shows the second joint acceleration will create inertial forces f_{2CY}, f_{3CY}, and inertia moment, m_{2CY} and m_{3CY} on link 2 and 3, respectively. A positive joint 2 acceleration causes positive f_{2CX} and f_{3CX}, which would increase m_{1Y}, i.e. increasing TOM and making the system less stable. A negative joint 2 acceleration causes negative f_{2CX} and f_{3CX}, which would decrease m_{1Y}, i.e. reducing TOM and making the system more stable. Figure 9 (c) shows the inertial forces caused by joint 3 acceleration.

To explain further, Figure 9 (d) illustrates the force situation for all three joint accelerations. The first joint acceleration will create an inertial force f_{1CY} and an inertial moment m_{1CZ} on link 1. The first two joint accelerations will create an inertial force $\mathbf{f}_{2C}$ and an inertial moment $\mathbf{m}_{2C}$ on link 2. All three joint accelerations will create an inertial force $\mathbf{f}_{3C}$ and an inertial moment $\mathbf{m}_{3C}$ on link 3, both having the same effects as that of link 2. These forces are generated by three absolute angular accelerations, α_1, α_2, and α_3. α_1 only has a component in either positive or negative Z direction and creates an inertial force and an inertial moment on all

Figure 9. Forces and moments derived from joint acceleration



links causing asymmetry. α_2 has an additional component in Y direction. If this acceleration component is in the positive direction, it generates an inertial force component in the positive X direction and an inertial moment in the positive Y direction on link 2, which will increase TOM. If this acceleration component is in the negative direction, it creates an inertial force component in the negative X direction and an inertial moment in the negative Y direction, which will reduce TOM. The effect of α_3 is similar to that of α_2. As mentioned before, increasing TOM makes the system less stable, whereas reducing TOM makes the system more stable.

SLIP ANALYSIS

Slip Criterion

As mentioned in Section 2, the force components f_{PX}, f_{PY} and moment component m_{PZ}, are in-plane components that would cause system to slip. In practice, prior to slip analysis, the tip-over index should be used first to ensure that the system is stable.

In Figure 2, total wrench $\mathbf{w}_{\mathrm{PF}}$ combines all the wrenches and acts on point O_{P}. It has three force components $\mathbf{f}_{\mathrm{PF}} = \left[\mathrm{f}_{\mathrm{PX}},\ \mathrm{f}_{\mathrm{PY}},\ \mathrm{f}_{\mathrm{PZ}}\right]^{\mathrm{T}}$ and three moment components $\mathbf{m}_{\mathrm{PF}} = \left[\mathrm{m}_{\mathrm{PX}},\ \mathrm{m}_{\mathrm{PY}},\ \mathrm{m}_{\mathrm{PZ}}\right]^{\mathrm{T}}$, i.e. $\mathbf{w}_{\mathrm{PF}} = \left[\mathbf{f}_{\mathrm{PF}}{}^{\mathrm{T}},\ \mathbf{m}_{\mathrm{PF}}{}^{\mathrm{T}}\right]^{\mathrm{T}}$. $\mathbf{w}_{\mathrm{PF}}$ is expressed as

$$\mathbf{f}_{\mathrm{PF}} = \mathbf{f}_{\mathrm{M}} + \mathbf{g}_{\mathrm{A}} + \mathbf{g}_{\mathrm{P}} \tag{47}$$

$$\mathbf{m}_{\mathrm{PF}} = \mathbf{m}_{\mathrm{M}} + \mathbf{f}_{\mathrm{M}} \times \mathbf{d}_{\mathrm{M}} + \mathbf{g}_{\mathrm{A}} \times \mathbf{d}_{\mathrm{A}} \tag{48}$$

$\mathrm{n}_{\mathrm{support\_i}}$ is denoted as the support force at the $i\mathrm{th}$ wheel in Z_{P} direction. Under the precondition that the system is stable, $\mathrm{n}_{\mathrm{support\_i}}$ always exists, meaning non-negative. The summation of all the wheel support forces can be related to f_{PZ} as:

$$\mathrm{f}_{\mathrm{PZ}} = \sum \mathrm{n}_{\mathrm{support\_i}} \tag{49}$$

Mathematically, the problem of slip can be presented as follows. In order to prevent slip, the support forces of the wheels must meet a friction force constraint given below:

$$\sqrt{\mathrm{f_{PX}}^2 + \mathrm{f_{PY}}^2} \leq \mu \mathrm{f_{PZ}} \tag{50}$$

where μ is the kinetic friction coefficient μ_{k} or static friction coefficient μ_{s}, depending on the working condition.

Eqn. (50) represents a formulation for the friction cone. To relate this cone to the friction forces in X and Y direction, most researchers replace it by the inscribed pyramid as a conservative method or a circumscribed pyramid as a less conservative one (Komatsu, Endo, & Hodoshima, 2013; Mahapatra, Toy, & Bhavanibhatla, 2015). In this study, a conservative method is chosen as the friction constraint expressed by the following inequalities:

$$\mathrm{f_{PX}} < \mu' \cdot \sum \mathrm{n_{support\_i}} \tag{51}$$

$$\mathrm{f_{PY}} < \mu' \cdot \sum \mathrm{n_{support\_i}} \tag{52}$$

where $\mu' = \sqrt{2}\mu \,/\, 2$.

Furthermore, a friction moment constraint must be met as well to prevent spinning, which is expressed as:

$$\mathrm{m_{PZ}} \leq \mu' \cdot \mathrm{r_i} \cdot \sum \mathrm{n_{support\_i}} \tag{53}$$

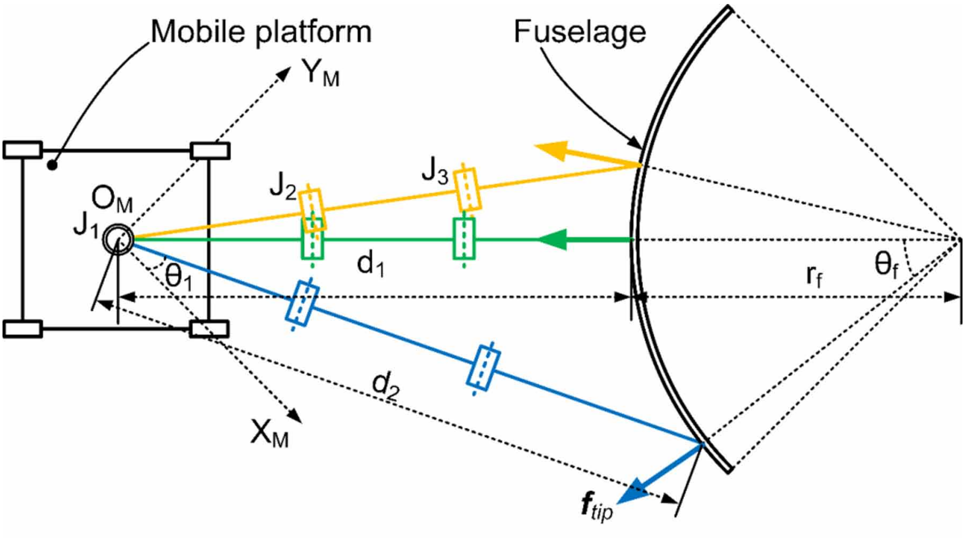
Where $\mathrm{r_i}$ is the distance from the i th wheel to the origin point $\mathrm{O_P}$.

Eqns. (51-53) are the criteria for mobile manipulator's slip analysis (Song, Xi, Guo, Tu, & Li, 2017a). The components $\mathrm{f_{PX}}$, $\mathrm{f_{PY}}$ and $\mathrm{m_{PZ}}$ can be obtained by eqn. (47-48). $\mathbf{f}_{\mathrm{M}}$ and $\mathbf{m}_{\mathrm{M}}$ in these two equations depend on the dynamics effect of manipulator that will be studied in the following section.

Tip Force Case

The effect of the tip force on slip condition is investigated by examining the stand-off distance. This distance $\mathrm{d_1}$ as shown in Figure 10 represents the distance between

Figure 10. The stand-off distance in reaction force case



the mobile manipulator and a target, in this case a fuselage. The mobile platform must hold still in order for the manipulator to perform a task, such as riveting on fuselage.

In most operations such as fuselage riveting, the tool axis should always align with the surface normal of the workpiece which can be determined through laser scanning (Song, Xi, Guo, Ming, & Lin, 2015). In other words, the direction of tip force is also in line with the surface normal of the workpiece. This direction is represented by angle θ_f, only depending on the riveting position. θ_f is fixed when the riveting position is fixed. As mentioned before, the manipulator has the last three joints to control orientation to follow the normal direction. The riveting distance d_2 shown in Figure 10 for different riveting positions is expressed as

$$d_2 = \text{sqrt}\left(\left(d_1 + r_f\right)^2 + \left(r_f\right)^2 - 2\left(r_f + d_1\right) \cdot r_f \cos\left(\theta_f\right)\right) \tag{54}$$

where r_f is the radius of fuselage. For a fixed riveting position, the slip condition of system mainly depends on the stand-off distance, as d_2 is a function of d_1.

Now let us look at forces. Based on a Newton-Euler formulation, the constraint wrench $\mathbf{w}_M$ in this case can be expressed as

$$\mathbf{w}_{\mathrm{M}} = {}^{\mathrm{s}}\mathbf{w}_{\mathrm{g1}} + \mathbf{w}_{\mathrm{p1}} + \mathbf{w}_{\mathrm{rf1}} \tag{55}$$

In eqn. (55), the first term ${}^{\mathrm{s}}\mathbf{w}_{\mathrm{g1}}$ is due to link weight, and it is only affected by the configuration of the links which depends on the stand-off distance. ${}^{\mathrm{s}}\mathbf{w}_{\mathrm{g1}}$ has a non-zero force component in Z_{M} direction and non-zero moment components in X_{M} and Y_{M} directions, with all other components zero. Non-zeroforce component in Z_{M} direction is equal to the gravity force of manipulator which remains constant. Non-zero moment components in X_{M} and Y_{M} directions will affect the tip-over stability but will not contribute to the slip condition of system.

The second term $\mathbf{w}_{\mathrm{p1}}$ is due to the tip payload $\mathbf{w}_{\mathrm{p(n+1)}}$ transferred from the tip to the base. $\mathbf{w}_{\mathrm{p(n+1)}}$ only has non-zero force component in Z_{M} direction with all other force and moment components zero. When this force component is transferred to joint 1, there are additional moment components in X_{M} and Y_{M} directions, but these moment components will not contribute to slip. Based on eqns. (47) and (48), f_{PX}, f_{PY} and m_{PZ} are all zero since f_{MX}, f_{MY} and m_{MZ} are all zero. In other words, $\mathbf{w}_{\mathrm{p1}}$ does not contribute to the forces/moments that would cause slip.

The last term $\mathbf{w}_{\mathrm{rf1}}$ is due to the tip applied wrench transferred to the manipulator base. For riveting, the tip force is from the tool, while the tip moment is a zero vector since there is no additional moment in the riveting process. $\mathbf{w}_{\mathrm{rf1}}$ does contribute to the forces/moments that would cause slip.

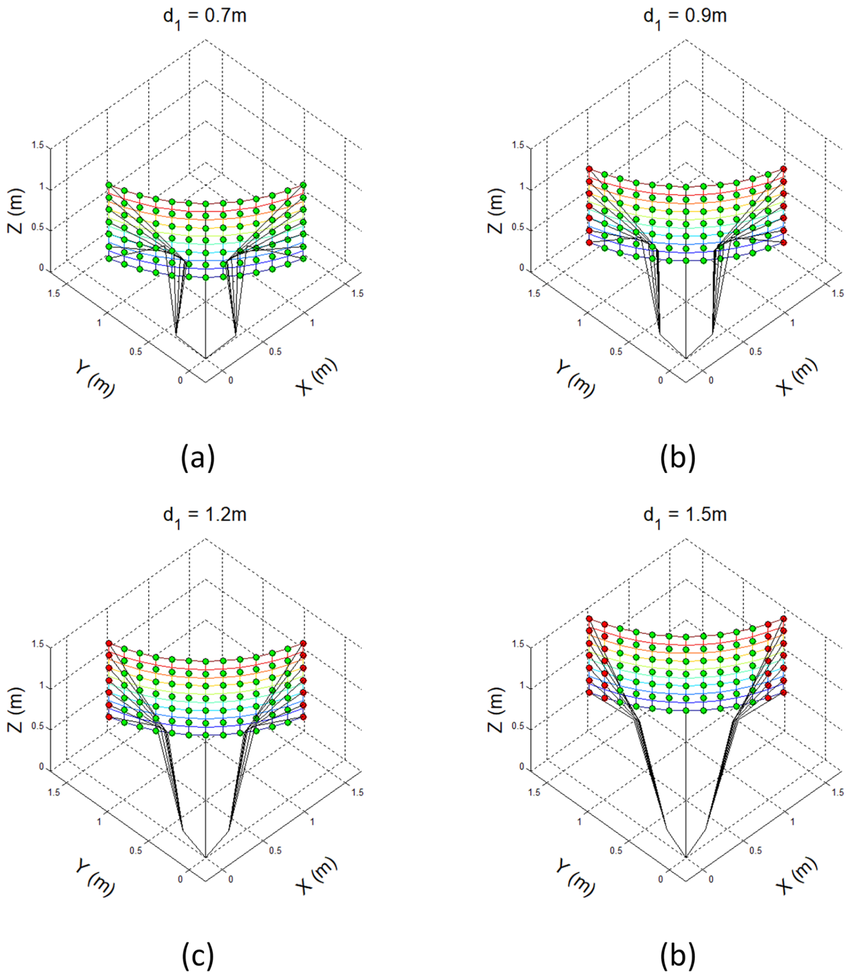
It was found in (Qu, Shi, Dong, Feng, Zhu, & Ke, 2014) that the impact force of percussive riveting is ranging from 800 to $1000\mathrm{N}$ under an input pneumatic pressure of $0.6\mathrm{MPa}$ fuselage riveting. In the following analysis, the tip force $\left|\mathbf{f}_{\mathrm{tip}}\right|$ is set as $1000\mathrm{N}$, expressed as

$$\mathbf{f}_{\mathrm{tip}} = 1000 \cdot \left[\cos\left(,_{\mathrm{f}}\right), \sin\left(,_{\mathrm{f}}\right), 0\right]^{\mathrm{T}} \tag{56}$$

As mentioned before that θ_{f} represents the riveter direction that is in line with the surface normal at each riveting position.

The flowchart of computing no slip range of stand-off distance is as follow. By following a standard rivet pattern, the wrench from manipulator onto mobile platform is computed at different rivet positions. Using the criteria in eqns. (51-53), the slip condition can be computed and the no slip range of stand-off distance can be obtained. All rivet position will be checked and increasing d_1 is to obtain last stand-off distance.

Figure 11. The simulation results with different d_1 *(a)* $\mathrm{d}_1 = 0.7\mathrm{m}$*, (b)* $\mathrm{d}_1 = 0.9\mathrm{m}$*, (c)* $\mathrm{d}_1 = 1.2\mathrm{m}$*, (d)* $\mathrm{d}_1 = 1.5\mathrm{m}$



(a)

(b)

(c)

(b)

To this end, it can be seen that when the manipulator performs riveting while the mobile platform holds still, the only wrench could affect slip is $\mathbf{w}_{\mathrm{rf1}}$. This term is mainly affected by the stand-off distance d_1. Figure 11 shows this effect on slip problem through simulation using the formulation presented in Section 3. In Figure 11, the curved surface represents a fuselage, and the black solid lines represent links of the manipulator. The green points on the fuselage indicate where the system does not slip when the manipulator rivets at this point, while the red points indicate the positions where the system slips. It can be observed from the four figures that the system does not slip at all rivet points when d_1 is equal to $0.7\mathrm{m}$ (Figure 11 (a)). With the increase of d_1, the system begins to slip at the rivet points on both sides of the fuselage (Figure 11 (b) and (c)). When d_1 is equal to $1.5\mathrm{m}$, the slip locations are doubled.

This phenomenon can be easily understood. In this case, f_{PX} and f_{PY} depend on $\mathbf{f}_{\mathrm{tip}}$ only, but m_{PZ} depends on $\mathbf{f}_{\mathrm{tip}}$ and d_1. $\mathbf{f}_{\mathrm{tip}}$ is fixed for a given rivet position, so are f_{PX} and f_{PY}. However, with the increase of d_1, m_{PZ} increases to the point when the constraint in eqn. (53) cannot be satisfied, hence causing the system to slip.

It should be noted that though the impact force from the riveting process does not last long, the slip resulting from this force will still lead to the loss of posture accuracy. There are two solutions to this problem. One is to decrease stand-off distance, and another is to decrease θ_{f}. Both methods require to place the mobile manipulator properly with respect to the workpiece. In the actual riveting, these two methods work equally well.

It should be noted as well that the height of manipulator's tip in this case does not affect slip. The height changes the moment components of $\mathbf{w}_{\mathrm{p1}}$ and $\mathbf{w}_{\mathrm{rf1}}$ in X_{M} or Y_{M} direction, and these two components will only cause the tip-over problem of the system, but not slip. The simulation results shown in Figure 11 also proved this principle. The slip condition is not relevant to the height of the robot tip position.

Therefore, it can be concluded that as far as the tip force is concerned, the system will slip when the mobile manipulator is located at a long stand-off distance from the fuselage. By shortening this distance, the system will not slip.

Joint Speed Case

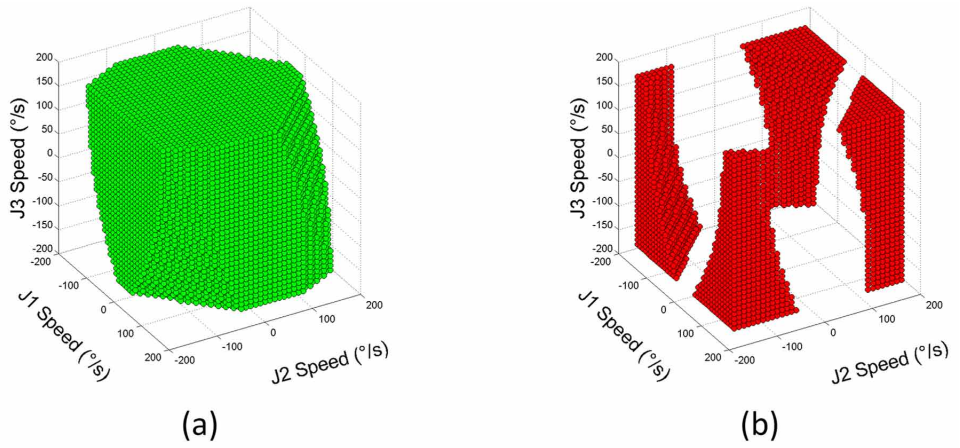
The effect of joint speeds on slip condition is investigated by examining the coupling term due to centrifugal forces and gyroscopic moments. In this analysis, all joint accelerations are set to zero. The constraint wrench in this case can be determined as eqn. (33).

The last two terms, ${}^{\mathrm{v}}\mathbf{w}_{\mathrm{C1}}$ and ${}^{\mathrm{v}}\mathbf{w}_{\mathrm{v1}}$ are obtained by accumulating the coupling term of each link through the corresponding wrench transformations to the manipulator base. The horizontal parts of the centrifugal forces will contribute to f_{1X} and/or f_{1Y}, and they will also contribute to m_{1Z} of ${}^{\mathrm{v}}\mathbf{w}_1$. Eventually they will contribute to slip problem. In addition, the gyroscopic moments will also contribute to m_{1Z}. As mentioned previously, the force components in X_{M} and Y_{M} directions and moment component in Z_M direction will contribute to slip problem while other components will not.

In our slip analysis, all joint speeds are considered for both directions. Simulations are carried out using the formulation presented. Figure 12 shows the results with the first three joint speeds. The speed of each joint ranges from $-175^\circ / \mathrm{s}$ to $175^\circ / \mathrm{s}$. The points in green color in Figure 12 (a) represent the corresponding joint speeds at which the mobile manipulator will not slip, while the points in red color in Figure 12 (b) represent the joint speeds at which the mobile manipulator will slip.

It can be observed from Figure 12 (b) that the red points mainly gather in the four edges and four corners. The four edges represent that slip occurs when joint 1 rotates at high speed while joint 2 is also rotating. Based on this, it is reasonable to predict that with the increase of joint 1 speed, the system may slip even though joint 2 or 3 does not rotate. The reason for this is that the centrifugal forces in the horizontal direction caused by joint 1 speed on all links and payload will contribute to f_{1X} and f_{1Y} of ${}^{v}\mathbf{w}_1$ in eqn. (33), eventually leading to slip of system. Furthermore, the four corners represent that slip occurs when joint 1, 2 and 3 all rotate at high speed with the last two joint rotations in the same direction. In other words, the system will most likely to slip if joint 1 rotates at high speed coupled with joint 2 and 3 rotating in the same direction at high speed. The reason for this is that since joints 2 and 3 are parallel, their speeds in the same direction will increase the absolute angular velocity of link 3, hence increasing the centrifugal force and gyroscopic moment. It should be noted that without joint 1 rotation, joint 2 and 3 rotations alone will not cause slip.

Figure 12. Slip condition for joint speed case



As far as fuselage riveting process is concerned, three directions are commonly used to move the manipulator to follow a rivet pattern. The first one is to move the tool above or below the current position in a vertical plane. The second one is to move the tool to the right or left of the current position in a horizontal plane. The third one is to move the tool in an oblique direction. When the manipulator moves in the first direction, the two parallel joint 2 and 3 usually rotate in the same direction. When the manipulator moves in the second direction, joint 1 usually rotates but joint 2 and 3 rotate at a slow speed that can be ignored. When the manipulator moves in the third direction, all three joints rotate. The following conclusions can be drawn based on the above joint speed simulation results. The system will not likely slip when the manipulator moves in the first direction. When the manipulator moves in the second direction, the system will not likely slip with a low joint 1 speed but likely slip with a fast joint 1 speed. The system will most likely slip when the system moves in the third direction.

Joint Acceleration Case

The effect of joint accelerations on slip condition is investigated by examining the inertia forces and moments. In this section, all joint speeds are set to zero and the constraint wrench is determined by eqn. (42).

Simulations are carried out using the formulation presented in the preceding section. Figure 13 (a) shows the wrench for joint 1 acceleration. This joint acceleration will create the inertia forces and moments on all links and payload. The joint 1 acceleration contributes a great deal to system's slip because the inertia forces and moments caused by joint 1 acceleration is in $X_M - Y_M$ plane and will contribute directly to f_{1X} and/or f_{1Y} of ${}^{a}\mathbf{w}_1$ in (42).

Figure 13. Forces and moments derived from joint acceleration

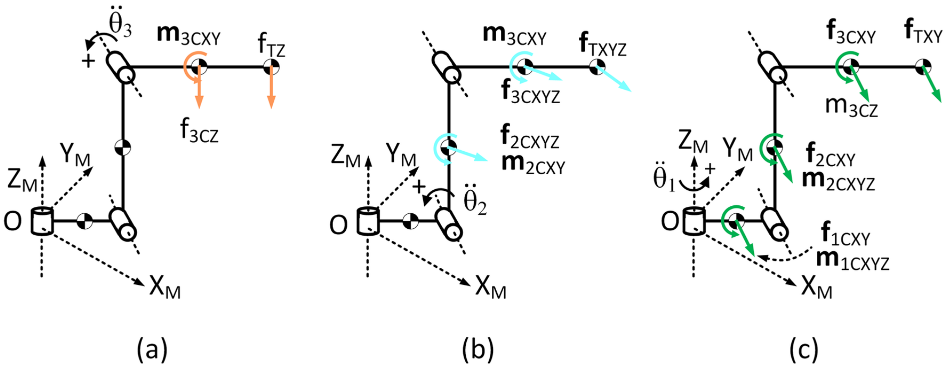


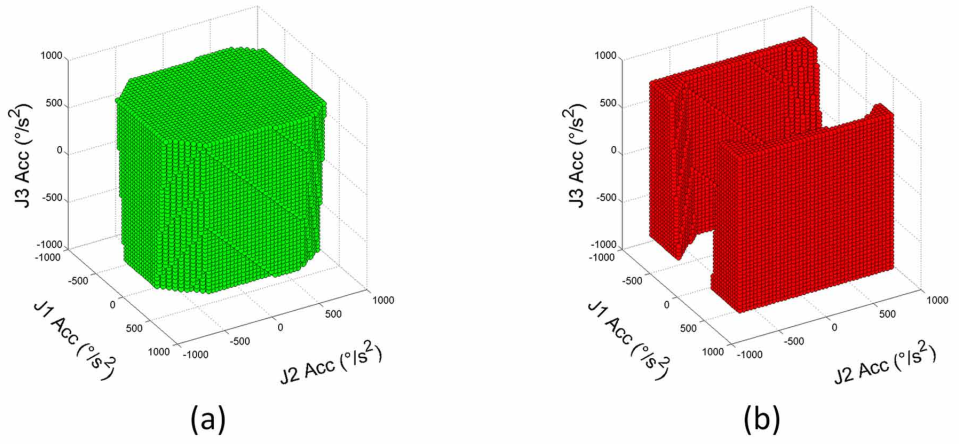
Figure 13 (b) shows the wrench for joint 2 acceleration. It will create the inertial forces and moments in vertical plane on link 2 and 3 and payload, respectively. They contribute to f_{1X} and f_{1Y}, thus to system slip. As shown in Figure 13 (c), joint 3 acceleration would yield a different result. The inertia force and moment caused by joint 3 acceleration which is in Z_{M} direction will contribute little to the support force compared to the gravitational force of the entire system, so it will have little effect on system slip condition.

Furthermore, the first three joint accelerations are considered to obtain the slip condition. Figure 14 shows the result with the acceleration of each joint ranging from $-800° / \mathrm{s}^2$ to $800° / \mathrm{s}^2$. Though in most cases the acceleration will not reach the maximum values, sudden braking will cause a high acceleration within a very short period of time. In Figure 14 (a) the points in green color represent the joint accelerations at which the mobile manipulator will not slip, whereas the points in red color in Figure 14 (b) represent the accelerations at which the mobile manipulator will slip.

By examining the distribution of the red points, it can be seen that they gather on two sides, meaning slip occurs when joint 1 is at high acceleration in either direction. Thus, it is also reasonable to state that joint 2 acceleration will affect slip a little and joint 3 acceleration will not affect slip at all.

Once joint 1 acceleration reaches beyond a certain value, the system always slips regardless of the other two joint accelerations. In actual riveting, it means that sudden braking of joint 1 will cause system to slip. Sudden braking occurs in a very short period of time, so does slip. Frequent sudden braking will lead to frequent slip. Eventually this slip will cause the loss of position accuracy. Especially the

Figure 14. Slip condition for joint acceleration case



manipulator will repeat the same process from one line riveting to another line, sudden braking will occur many times, leading to the loss of accuracy at end. One solution to this problem is to plan the trajectory to avoid sudden braking. In other words, the system can decelerate before braking. Another solution to this problem is to plan a riveting trajectory that can avoid joint 1 acceleration as much as possible.

As analyzed above, there are three directions in which the manipulator moves for riveting. Based on the above joint acceleration simulation result, it is reasonable to state that the system will likely slip when the manipulator moves with large joint 1 acceleration. The system will not likely slip when the manipulator moves in the first direction, i.e. to the above or below of the current position in a vertical plane, since there is no joint 1 acceleration. The system will likely slip when the manipulator moves in a horizontal plane or in an oblique direction, in which case, the joint accelerations should be limited.

Experiment

The slip analysis presented in this study is validated using the mobile manipulator developed by Shanghai University. For the experiment an industrial camera is installed in the front of the mobile manipulator facing a calibrated panel. The slip displacement can be measured with respect to the calibration panel.

The slip displacement was measured on different paths. On all paths of experiment, the last three joints, i.e. joint 4 to 6 hold still, so the wrist is considered as part of the payload. On path 1, joint 1 holds still, while joint 2 and 3 rotate in the same direction. On path 2, joint 1 rotates while joint 2 and 3 hold still. On path 3, joint 1 rotates while joint 2 and joint 3 rotate in the same direction. These three paths respectively correspond to the three different directions. Path 1 represents that the manipulator moves to above or below the current position in a vertical plane. Path 2 represents that the manipulator moves to the left or right of the current position in a horizontal plane. Path 3 represents that the manipulator moves in an oblique direction.

In experiment, the manipulator moved from joint start positions (Position A) to stop positions (Position B). The slip effect resulting from the different joint speeds was measured. Four velocities were chosen for the experiment: v3000, v4000, v5000 and v6000 respectively. All of them are linear velocities for the tool center point, and the unit is mm/s (ABB Robotics, n.d.). A large linear velocity means a large joint speed, which usually means a large acceleration.

The procedure of experiment can be stated as follows. Before manipulator's movement, the position of the mobile manipulator was measured using a vision calibration unit. Then the manipulator moved along a planned path from a start position to a stop position at the selected speed and then moved back with a small velocity of v500 at which the system does not slip. After ten cycles, the manipulator stopped and the new position of the mobile manipulator was measured again. The slip displacement can be obtained by looking at the difference between the two position measurements.

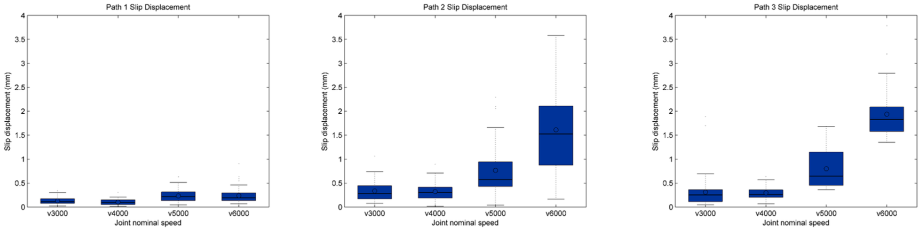
The experiment results of the three paths with the four joint speeds were obtained. Figure 15 (a) to (c) show the slip displacement results for path 1 to 3, respectively. For path 1, when joint 2 and 3 rotated in the same direction, it can be found that tiny slip occurred. There was no much difference between four joint velocities on this path. Taking v5000 as an example, for 50 slip displacement data, the median and mean are 0.218 mm and 0.233mm. As analyzed before, only speeds and accelerations of joint 2 and 3 do not cause slip of the system. The experiment confirms the theoretical analysis. It means that the manipulator can move to the above or below of the current position in a vertical plane with a large velocity and acceleration with no slip possibility. Based on the experiment result, considering both accuracy and efficiency requirements, the velocity chosen can be up to v6000 for a path in the real riveting.

For path 2, the slip displacement is not obvious when the speed is smaller than v5000, but it increases significantly when the speed is larger than v5000. The median and mean are 0.573mm and 0.765mm for v5000, while they are 1.524mm and 1.613mm for v6000. As analyzed in the previous sections, the horizontal forces caused by large joint 1 speed and acceleration will likely lead the system to slip. The experiment confirmed this theoretical analysis. Based on the experiment results it can be seen that when the manipulator moves in the horizontal plane, joint 1 speed should be limited to v4000 in order to avoid the slip. The joint 1 acceleration caused by sudden braking should also be paid attention to.

For path 3, it is the combination of path 1 and path 2. This combination will cause the gyroscopic moment and increase the tip velocity and acceleration, eventually leading the system even more likely to slip. When the speed is v6000, the median and mean of slip are 1.829mm and 1.937mm, respectively. They are both larger than those of path 2 under the same speed. The experiment result shows that the system is most likely to slip when joint 2 and 3 rotate in the same direction at high speed along with joint 1 rotation. The experiment confirmed this theoretical analysis. Based on the result, the velocity should be limited to v4000 to reduce the possibility of slip in the real riveting if the manipulator moves in an oblique direction. The sudden braking should also be paid attention to.

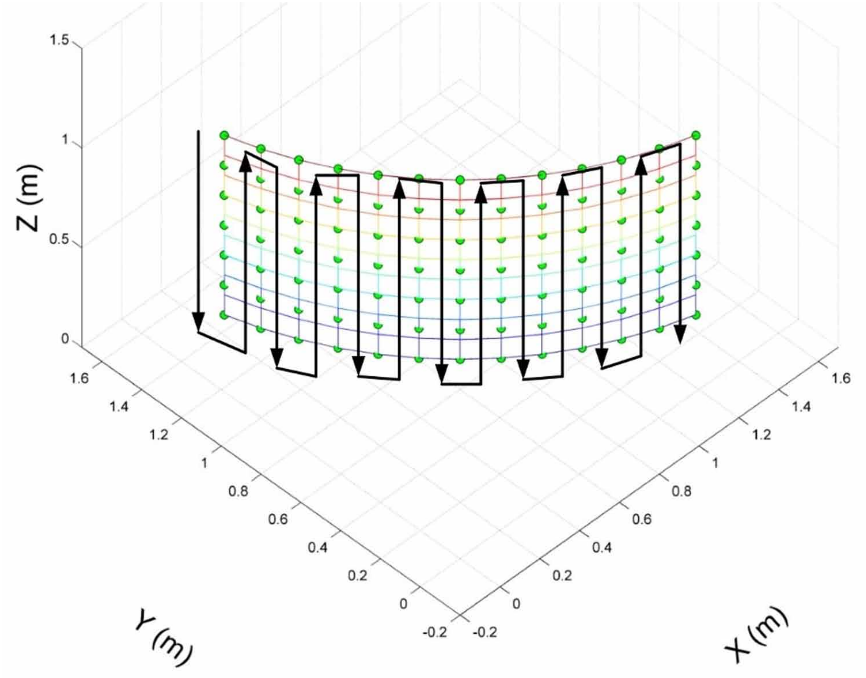
Based on the simulations and experiments, in our riveting process, the stand-off distance is assigned as 0.7m and the optimum rivet consequence is determined as shown in Figure 16. With these selections, the slip of system caused by the dynamics effect of manipulator is minimized and the accuracy of riveting can be guaranteed.

Figure 15. Slip displacement



(a) (b) (c)

Figure 16. The optimum riveting sequence



DESIGN BASED ON TIP-OVER STABILITY

Optimization Problem Formulation

As mentioned before, the negative value of TOM represents a stable tip-over margin. For tip-over evaluation, eqn. (20) will be applied to cover all the tip-over axes formed by a combination of all adjacent wheels. Symbolically, TOM can be written as the following function

$$\mathrm{TOM} = \mathrm{f}\left(\mathrm{n}, \mathrm{r}, \mathrm{m}_{\mathrm{P}},\ \mathbf{d}_{\mathrm{M}}, \mathbf{d}_{\mathrm{A}}\right) \tag{57}$$

Eqn. (57) states that the tip-over moment is affected by the size of the mobile platform denoted by r, the number of wheels denoted by n, the mass of the mobile platform denoted by m_{P}; and the placements of the manipulator and its accessory, denoted by $\mathbf{d}_{\mathrm{M}}$ and $\mathbf{d}_{\mathrm{A}}$, respectively. All these variables are taken as the parameters in our proposed optimization method.

For a selected manipulator, its workspace volume is given, denoted by v_{W}. The goal of our optimization is to achieve a maximum tip-over stability region denoted by v_{S} over the manipulator's workspace. A stable region ratio (SRR) is defined as $\mathrm{SRR} = \mathrm{v}_{\mathrm{S}} / \mathrm{v}_{\mathrm{W}}$, which normalizes v_{S} against v_{W}. As a result, maximum v_{S} is always relative to given v_{W}. While attempting to maximize SRR, it is also intended to minimize TOM. As mentioned before, a larger negative value of TOM means a higher stable margin. As indicated in eqn. (20), TOM is affected by the reaction wrench which is a function of manipulator's pose and motion. Hence, the objective function of our optimization problem may be defined as a max-min optimization problem:

$$\arg \max_{\theta \in \mathrm{v}_{\mathrm{W}}} \min \left\{\mathrm{SRR}\left(\mathrm{TOM}\left(\theta\right)\right)\right\} \tag{58}$$

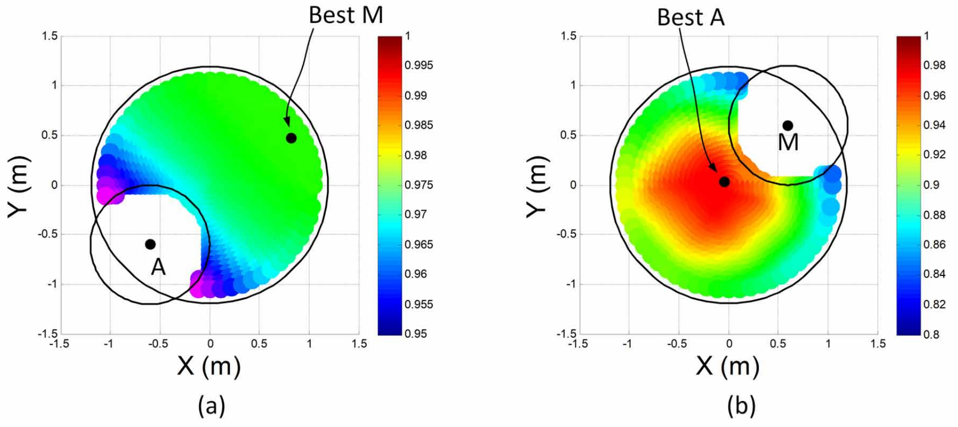
where θ is the vector of joint angles of the manipulator which are used to search over the workspace. The optimization problem given in eqn. (58) is subject to the following constraints.

Placement of the Manipulator and Accessory

As explained in Figure 2, the placement of the manipulator and the accessory is represented by the eccentric distance $\mathbf{d}_{\mathrm{M}}$ and $\mathbf{d}_{\mathrm{A}}$, respectively. For this investigation, the mass and size of the mobile platform, m_{P}, r are pre-selected using the highest allowable power and size, but a common four wheels design is adopted. Figure 17 shows TOM results of two cases. Case 1 $\mathbf{d}_{\mathrm{A}}$ is fixed but $\mathbf{d}_{\mathrm{M}}$ changes, Case 2 is vice versa.

Figure 17. (a) shows the simulation result for Case 1. In this figure, the large circle represents the mobile platform and the small circle represents a separation area between the manipulator and the accessory where the manipulator cannot be

Figure 17. SRR *simulation result with different* $\mathbf{d}_{\mathrm{M}}$ *and* $\mathbf{d}_{\mathrm{A}}$



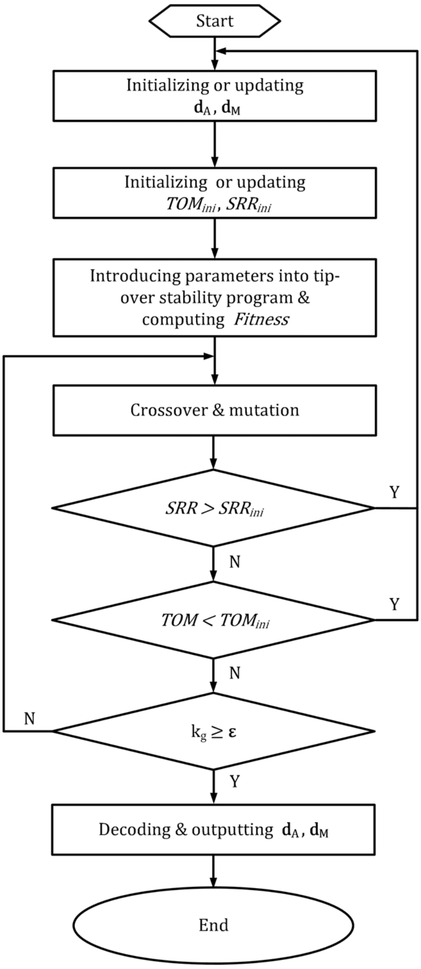
installed. The color scale represents the SRR value and 1 corresponds to 100%. It can be seen from this figure that the mounting position of the manipulator on the mobile platform plays an important role on tip-over stability of the system. An appropriate $\mathbf{d}_{\mathrm{M}}$ can make the system more stable with a higher SRR, while an inappropriate one makes the system less stable with a lower SRR. In this search, the best $\mathbf{d}_{\mathrm{M}}$ is $\left[0.8227,\, 0.4750,\, 0\right]\mathrm{m}$, with SRR of 96.91%, but the highest TOM in the remaining unstable region is $1345.3\ \mathrm{Nm}$. Figure 17 (b) shows the simulation results for Case 2. It can be seen that the mounting position of the accessory also has some effect. An appropriate $\mathbf{d}_{\mathrm{A}}$ can make the system more stable with a higher SRR while an inappropriate one makes the system less stable with a lower SRR. In this search, the best $\mathbf{d}_{\mathrm{A}}$ is $\left[-0.0372,\, 0.0335,\, 0\right]\mathrm{m}$ with SRR of 100% and TOM of $-1.4897\ \mathrm{Nm}$.

Optimization for Largest Tip-Over Stability Margin

In this section, instead of grid-based search, Genetic Algorithms (GA) are adopted to determine the optimum values of $\mathbf{d}_{\mathrm{A}}$ and $\mathbf{d}_{\mathrm{M}}$ (Song, Xi, Guo, & Lin, 2016). Two strategies are considered. The first one strives at achieving the largest tip-over stability margin, i.e. the most negative value of TOM. The second strategy strives at achieving the largest distance of $\mathbf{d}_{\mathrm{M}}$. Both strategies guarantee 100% SRR.

The steps of the first optimization are shown in Figure 18 and given as follows:

Figure 18. Flow-chart for largest tip-over stability margin



1. Set $\mathbf{d}_A$ and $\mathbf{d}_M$ initial population at $\mathbf{d}_A = \left[-0.6, -0.6, 0\right]\mathrm{m}$ and $\mathbf{d}_M = \left[0.6, 0.6, 0\right]\mathrm{m}$, based on the grid-based search method. Considering the actual application, the manipulator will be always mounted along the central axis of the mobile platform in the forward direction. However, the position of the accessory, i.e. $\mathbf{d}_A$ can be searched all around the mobile platform. The initial stable region ratio $\mathrm{SRR}_{\mathrm{ini}}$ is 96.87%, and the initial tip-over moment $\mathrm{TOM}_{\mathrm{ini}}$ is $1339.8\ \mathrm{Nm}$. GA parameters are chosen as: the crossover fraction is 0.8, the size of the population is 25, and the maximum generation of iterations denoted by ε is 100.

2. Calculate TOM and SRR and update $\mathbf{d}_{\mathrm{A}}$ and $\mathbf{d}_{\mathrm{M}}$.
3. Continue step 2, till the termination condition is met, i.e. iterative generation $\mathrm{k}_{\mathrm{g}} \geq \varepsilon$.

The optimization is implemented using the MATLAB® GA toolbox. The optimized results are $\mathbf{d}_{\mathrm{A}} = \left[-0.126,\ -0.0169, 0\right]\mathrm{m}$ and $\mathbf{d}_{\mathrm{M}} = \left[0.2293, 0.2293, 0\right]\mathrm{m}$, which provide $\mathrm{SRR} = 100\%$ and $\mathrm{TOM} = -592.2\mathrm{Nm}$.

Optimization for Largest Distance of $\mathbf{d}_{\mathrm{M}}$

This optimization attempts to install the manipulator as close to the edge of the mobile platform as possible without loss of tip-over stability. This way, the manipulator's end-effector will have a greater accessibility. This strategy will still guarantee 100% SRR, but the tip-over margin is not as high as the first strategy.

Similar to Case 1, the manipulator is mounted along the central axis in the forward direction. The position of accessory i.e. $\mathbf{d}_{\mathrm{A}}$ can be searched all around the mobile platform. Figure 19 shows the flow chart. In this simulation, the results of Case 1 are taken as the initial population for this GA optimization. The termination condition of GA is that iterative generation $\mathrm{k}_{\mathrm{g}} \geq \varepsilon$ or the manipulator already reaches the edge of the mobile platform, i.e. $\left|\mathbf{d}_{\mathrm{M}}\right| = \mathrm{r} - \mathrm{r}_{\mathrm{M}}$. The simulation results in this case are $\mathbf{d}_{\mathrm{A}} = \left[-0.2764,\ -0.0165, 0\right]\mathrm{m}$ and $\mathbf{d}_{\mathrm{M}} = \left[0.7722, 0.7722, 0\right]\mathrm{m}$, with $\mathrm{SRR} = 100\%$ and $\mathrm{TOM}= -227.4\mathrm{Nm}$. Comparing the two cases, the second | $\mathbf{d}_{\mathrm{M}}$ | is 1.092m larger than the first | $\mathbf{d}_{\mathrm{M}}$ |= 0.3242m, while the second TOM is about half of the first.

APPLICATIONS AND CONCLUSIONS

The proposed analysis and optimization method has been applied to design two mobile manipulators, one for drilling/riveting for aerospace manufacturing as shown in Figure 20 (a). The selected manipulator is an ABB IRB 2600. The Mecanum wheels are selected for omni-direction moving. The prototype shown in Figure 20 (a) is optimized for largest distance of $\mathbf{d}_{\mathrm{M}}$. In actual use, the system's tip-over stability is well guaranteed.

Another mobile manipulator is also design with this method as shown in Figure 20 (b). This one is for baseline drawing for interior decoration. The selected manipulator

Figure 19. Flow-chart for largest distance of $\mathbf{d}_M$

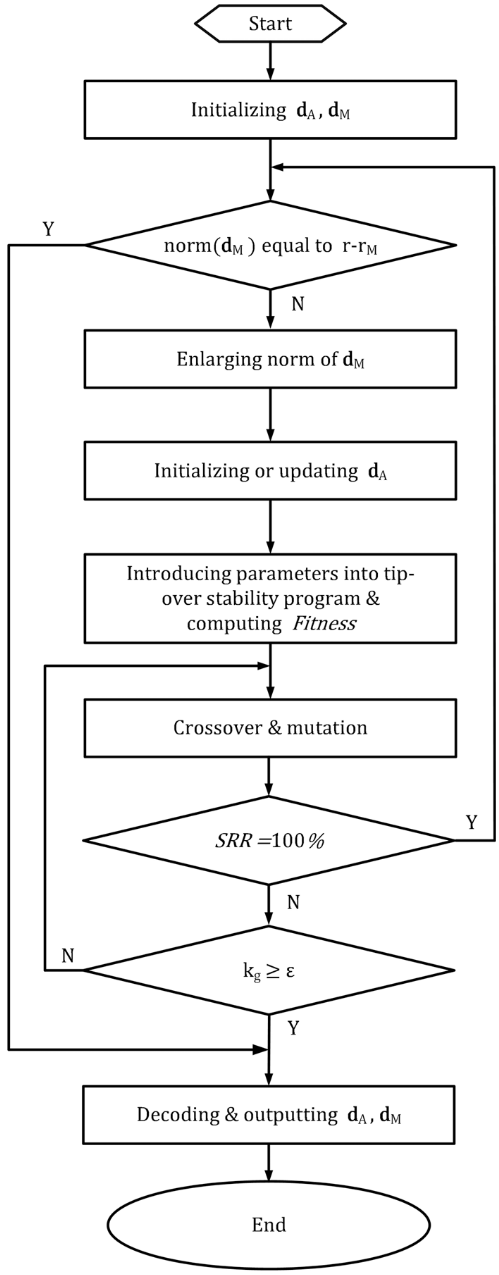
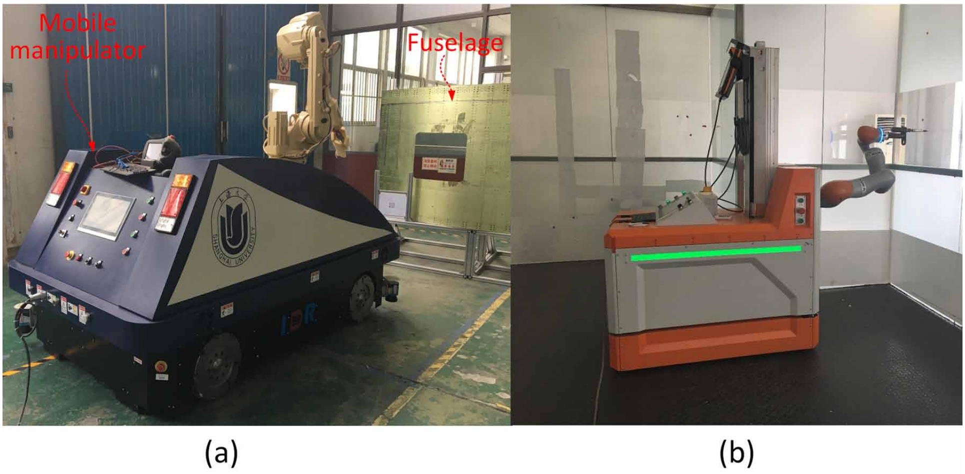

Start
Initializing $\mathbf{d}_A$, $\mathbf{d}_M$
norm($\mathbf{d}_M$) equal to r-r$_M$
Y
N
Enlarging norm of $\mathbf{d}_M$
Initializing or updating $\mathbf{d}_A$
Introducing parameters into tip-over stability program & computing *Fitness*
Crossover & mutation
SRR =100%
Y
N
$k_g \geq \varepsilon$
N
Y
Decoding & outputting $\mathbf{d}_A$, $\mathbf{d}_M$
End

Figure 20. Mobile manipulator for aerospace manufacturing (a) and for decoration engineering (b)



is KUKA IIWA. The Mecanum wheels are also selected for omni-direction moving. One linear elevator mechanism is also included into the design of system to cover the height of internal wall. This height factor is also considered for tip-over stability analysis and optimization.

According to the optimization and analysis, in actual operation, the manipulator moves with constraints obtained as results from different case analysis and consequently no tip-over occurs. For slip issue, the path is planned with joint speeds/accelerations restrained to decrease the slip phenomenon and ensure the accuracy of localization.

REFERENCES

Bianco, C. G. L. (2009). Evaluation of Generalized Force Derivatives by Means of a Recursive Newton–Euler Approach. *IEEE Transactions on Robotics*, *25*(4), 954–959. doi:10.1109/TRO.2009.2024787

Guo, S., Song, T., Xi, F. F., & Mohamed, R. P. (2017). Tip-Over Stability Analysis for a Wheeled Mobile Manipulator. ASME. J. Dyn. Sys. *Measurement and Control*, *139*(5), 054501–054501, 7. doi:10.1115/1.4035234

Komatsu, H., Endo, G., & Hodoshima, R. (2013). Basic Consideration About Optimal Control of a Quadruped Walking Robot During Slope Walking Motion. *IEEE Workshop on Advanced Robotics and Its Social Impacts (ARSO)*, 224–230. 10.1109/ARSO.2013.6705533

Lin, Y., Xi, F. F., Mohamed, R. P., & Tu, X. W. (2010). Calibration of Modular Reconfigurable Robots Based on a Hybrid Search Method. ASME. *Journal of Manufacturing Science and Engineering*, *132*(6), 061002–061002, 8. doi:10.1115/1.4002586

Liu, Y. G., & Liu, G. J. (2009). Modeling of tracked mobile manipulators with consideration of track–terrain and vehicle–manipulator interactions. *Robotics and Autonomous Systems*, *57*(11), 1065–1074. doi:10.1016/j.robot.2009.07.007

Liu, Y. G., & Liu, G. J. (2010). Interaction Analysis and Online Tip-Over Avoidance for a Reconfigurable Tracked Mobile Modular Manipulator Negotiating Slopes. *IEEE/ASME Transactions on Mechatronics*, *15*(4), 623–635. doi:10.1109/TMECH.2009.2031174

Mahapatra, A., Roy, S. S., & Bhavanibhatla, K. (2015). Energy-Efficient Inverse Dynamic Model of a Hexapod Robot. *International Conference on Robotics, Automation, Control and Embedded Systems (RACE)*, 1–7. 10.1109/RACE.2015.7097237

Qu, W. W., Shi, X., Dong, H. Y., Feng, P. J., Zhu, L. S., & Ke, Y. L. (2014). Simulation and Test on Process of Percussive Impact Riveting. *Journal of Zhejiang University (Engineering Science)*, *48*(8), 1411–1418.

Rehman, B. U., Focchi, M., & Lee, J. (2016). Towards a multi-legged mobile manipulator. *Robotics and Automation (ICRA), IEEE International Conference on. IEEE*, 3618-3624. 10.1109/ICRA.2016.7487545

ABB Robotics. (n.d.). *Technical reference manual: RAPID instructions, functions and data types*. ABB Robotics.

Song, T., Xi, F. F., Guo, S., & Lin, Y. (2016). Optimization of a Mobile Platform for a Wheeled Manipulator. ASME. *Journal of Mechanisms and Robotics*, *8*(6), 061007–061007, 14. doi:10.1115/1.4033855

Song, T., Xi, F. F., Guo, S., Ming, Z. F., & Lin, Y. (2015). A Comparison Study of Algorithms for Surface Normal Determination Based on Point Cloud Data. *Precision Engineering*, *39*, 47–55. doi:10.1016/j.precisioneng.2014.07.005

Song, T., Xi, F. F., Guo, S., Tu, X. W., & Li, X. H. (2017). Slip Analysis for a Wheeled Mobile Manipulator. ASME. J. Dyn. Sys. *Measurement and Control*, *140*(2), 021005–021005, 12. doi:10.1115/1.4037287

Vysin, M., & Knoflicek, R. (2003). The Hybrid Mobile Robot. *IEEE International Conference on Industrial Technology (ICIT)*, *1*(12), 262–264. doi: 10.1109/ICIT.2003.1290291

Xi, F. F. (2009). *Computational Dynamics (Graduate Course Lecture Notes)*. Toronto, ON, Canada: Ryerson University.

Chapter 3
Towards Multiple-Layer Self-Adaptations of Multi-Agent Organizations Using Reinforcement Learning

Xinjun Mao
National University of Defense Technology, China

Menggao Dong
National University of Defense Technology, China

Haibin Zhu
Nipissing University, Canada

ABSTRACT

This chapter proposes a multi-agent organization model for self-adaptive software to examine the autonomous components and their self-adaptation that can be occurred at either the fine-grain behavior layer of a software agent or the coarse-grain organization layer of the roles that the agent plays. The authors design two-layer self-adaptation mechanisms and combine them with reinforcement learning together to tackle the uncertainty issues of self-adaptation, which enables software agents to make decisions on self-adaptation by learning at run-time to deal with various unanticipated changes. The reinforcement learning algorithms supporting fine-grain and coarse-grain adaptation mechanisms are designed. In order to support the development of self-adaptive software, the software architecture for individual agents, the development process and the software framework are proposed. A sample is developed in detail to illustrate our method and experiments are conducted to evaluate the effectiveness and efficiency of the proposed approach.

DOI: 10.4018/978-1-5225-5276-5.ch003

INTRODUCTION

Our society highly depends on complex IT systems created by integrating and orchestrating independently managed systems. The complexity of such systems stems from the great number of the system's components, the dynamic relationships between them, and the continuous interactions between the system and its varying environment (Sommerville et al, 2012). Software systems in such a kind require self-adaptation in order to deal with unexpected internal and external changes and thereby to guarantee the flexibility, robustness, reliability of the system (Litoiu et al, 2017; Mahdavi-Hezavehi et al, 2017). Self-adaptive software is normally designed as a closed-loop system that aims to adjust various artifacts or attributes in response to changes in the 'self' and/or within the context (Cheng et al, 2009). Here, 'self' means the whole body of the software, mostly implemented in several layers, while the context encompasses everything in the operating environment (Salehie et al, 2009). Such systems are currently required in many application fields such as the CPS, enterprise, industry, (Camara et al, 2016; Camara et al, 2016)

Several challenges must be addressed when developing complex self-adaptive software, including abstraction, modeling, mechanism design, analysis method, software architecture, language and platform. In the past two decades, the software engineering community devotes to investigating innovative ways of developing, deploying, managing and evolving self-adaptive software systems (Cheng et al, 2009; Salehie et al, 2009). Various software engineering technologies for self-adaptive system have been proposed such as software architecture, control loop, running strategy and model, modelling and programming languages (Cheng et al, 2009; Camara et al, 2016; Weynes, 2017).

However, the evolution of software features and the increasing complexity of current software systems challenges existing software engineering technologies and at the same time pose several open issues to develop complex self-adaptive software.

First, many software systems over the Internet are ultra-large scale systems or coalition of systems-of-systems (Somerville et al, 2012; Weynes, 2017). Typically, these systems are composed of a great number of independent and autonomous components, and the whole systems are managed in a decentralized way. Therefore, traditional methods that assume the system can be controlled in a centralized way seems unsuitable. To investigate the self-adaption in such kind of systems and realize the software needs appropriate abstractions and models.

Second, self-adaptation in a software system can occur at different levels simultaneously to satisfy design objectives, ranging from low-level and fine-grain data and behaviors self-adaptation, to high-level and coarse-grain goal and responsibilities self-adaptation of an individual component, and even the structure-level adaptation managed by the whole systems. As the independence and decentralization of the individual software components of the software system, self-adaptations at different levels should be decided by corresponding components in an autonomous way. Therefore, questions must be answered as to the nature and types of self-adaptation at different levels and the mechanisms realizing these self-adaptations should be designed.

Third, for many complex systems that are evolving continuously and operate in open and dynamic environment, to precisely anticipate various changes and predefine the self-adaptation requirements at design-time is difficult and maybe impossible, because new and perhaps unknown system elements may dynamically enter or leave, unexpected events may occur. Actually, such systems are often built in an evolving way with varying, inherently conflicting, unknowable and diverse requirements (Somerville et al, 2012). Developers often lack complete knowledge of the system and its executing conditions before deployment, and how these uncertainties can be resolved at runtime (Weyns, 2017). Uncertainty in a self-adaptive system may be due to the simplifying assumptions, model drift, noise, parameters in future operation, and human in the loop, operating context, etc. (Esfahani et al, 20123; Mahdavi-Hezavehi et al, 2016). To tackle the uncertainty of self-adaptive system has become an open issue in the research and practice of self-adaptive software engineering. Obviously, traditional approaches to pre-defining the set of self-adaptation events and logic and encapsulating them into a software module at design-time by developers is infeasible.

To cope with the above challenges of self-adaptive software, we should seek suitable abstractions and models to represent the substantive characteristics of self-adaptive software and control its complexity, design practical mechanisms to support the decisions and running of self-adaptive software, and find effective approaches to tackling the uncertainties issues of self-adaptive software. From the design viewpoint, software engineering for self-adaptive systems should provide effective technologies to support the abstraction, decomposition, and organization to manage the complexity of self-adaptive software. On the other hand, from the deployment and running viewpoint, the designed self-adaptive system should not only control the self-adaptation operations of the system, but also be self-aware and context-aware, and more important, be capable of deciding their self-adaptation behaviors by themselves at run-time. We believe that the situatedness and autonomy of agent technology provide the potential to deal with the above issues, because situatedness enables software agents to perceive changes of the environment at run-time and autonomy enables them to make on-line self-adaptation decisions autonomously. Moreover, the multi-agent organization model of self-adaptive software is helpful to investigate various self-adaptation at different levels. Combining the self-adaptation mechanism with reinforcement learning method will provide a feasible way to enable self-adaptive systems to decide and take suitable operations to adjust their structure and behaviors in order to adapt to the uncertain changes occurring in themselves and the situated environment.

Against the above challenges of self-adaptive software, this paper proposes a multi-agent organization model for self-adaptive systems and multiple-layer self-adaptation mechanisms to examine the self-adaptation at different levels. The self-adaptation decision approach based on reinforcement learning are proposed to enable self-adaptive software to tackle uncertain changes. The software architecture, development process and software framework are also designed to support the development of self-adaptive software. The rest of this paper is organized as follows. First, we discuss the related works, which is followed by the introduction of the abstraction and model of self-adaptive systems. Then, we detail the mechanisms and algorithms for multiple-layer self-adaptations based on reinforcement learning. The following section describes the software architecture, development process and software framework of self-adaptive software with a case study, and experiment analysis and evaluation are made in the next section. Finally, conclusions and future works are discussed.

Related Works

In the past years, there are a great number of efforts and progresses in the field of software engineering for self-adaptive system, covering requirement, design, implementation, verification and validation (Cheng et al, 2009; Salehie et al, 2009). Weyns divided the research efforts on self-adaptive software into six waves, including automating tasks, architecture-based adaptation, runtime models, goal driven adaptation, guarantees under uncertainties, and control-based approaches. Each wave highlights a trend of interest in the research community (Weynes, 2017). Luca Sabatucci tried to give a meta-model to capture and explore the essential features of diverse self-adaptive software (Sabatucci et al, 2017).

In complex software systems like IoT (Bennacuer et al, 2016), self-adaptation can be performed at different levels and granularities such as architecture, component, parameter or data (Salehie et al, 2009), etc. The self-adaption in one level may trigger self-adaptation in another (Popescu et al, 2012). Most of existing research efforts on self-adaptation focus on some specific levels and seek engineering approaches to realizing the self-adaptation in term of software technologies, e.g., reflection, software architecture, control loop, etc. For example, architecture-based adaptation is mainly concerned with structural changes at the level of software architecture like adding and deleting components, or changing the connectors to adjust the structure and behavior of software system. Such approach is helpful to separate the concerns of the regular functionality of the system from the concerns that are subject of the adaptation (Weyns, 2017). Representative researches include Rainbow (Garlan et al, 2004), eARF (Abbas et al, 2016).

The investigations and practices of self-adaptive software engineering are tightly related with the paradigms of software and the technologies to implement the software. Multi-agent technologies are believed with the potentials to deal with the complexity of software system in term of high-level abstraction, problem decomposition and system organization solutions. Moreover, a software agent seems more suitable for constructing a self-adaptive system due to its two specific features and capabilities, one is the situatedness to perceive the changes of environment and the other is the autonomy to make decision on behaviors by itself. The self-adaptation of a multi-agent system can range from the data, parameters and behaviors of individual agents, to the structure (Lee et al, 2005) and even the organization of the whole system, which represents different levels of self-adaptation in a multi-agent system (Jensen et al, 2013). Typical researches are dynamic role assignments (Odell et al, 2003; Hilaire et al, 2002) and corresponding self-adaptation mechanisms (e.g., role-playing or enacting (Zhu et al, 2006; Mao et al, 2008; Viana et al, 2016), social rule or normative regulations (Viana et al, 2016; Cernuzzi et al, 2005)). Dynamic role playing or assignment (Odell et al, 2003; Hilaire et al, 2002; Zhu et al, 2006) are used as high-level abstractions and mechanisms to investigate the self-adaptive behaviors and analyze their characteristics at organization levels. Organization role is an important concept to examine the agent's behaviors and their dynamic. An agent in the organization can dynamically adjust its role through performance or assignment, depending on its organizational context. Several software engineering approaches for developing self-adaptive multi-agent system are proposed, e.g., software architecture of caste (Zhu et al, 2003), methodologies (Cernuzzi et al, 2005; Dastani et al, 2004), and even programming languages (Dastani et al, 2004; Zhu et al, 2003). However, most of the work assumes the self-adaptation requirements and the logic of multi-agent systems can be explicitly defined by developers at design-time.

Self-adaptive software should know how to decide its self-adaptation operations when some events occur. In existing researches, adaptation decisions can be made in static or dynamic ways. In the static option, the deciding process is hard-coded and its modification requires recompiling and redeployment of the system or some of its components. Developers can encapsulate the decision information as control strategies or programs by various strategies description languages or programming languages (Fredericks et al, 2016; Cheng e tal, 2009; Salehie et al, 2009). In dynamic decision-making on self-adaption, policies, rules of self-adaptation are externally defined and managed, so that they can be changed during run-time to satisfy new requirements (Cheng e tal, 2009; Salehie et al, 2009). Adaptation decisions can also be made in off-line or online ways. Off-line self-adaptation behaviors are actually made by developers at design-time, depending on pre-defined changes and adjustments. On-line self-adaptation decisions are made by software systems, which require autonomy and context-awareness. However, development of self-adaptive system should trade-off the timeliness and optimality provided by various self-adaptation mechanisms and decision methods. For example, (Pandey et al, 2016) presents a hybrid approach to planner, which combines deterministic planning with Markov Decision Process (MDP) planning to obtain the best of both worlds.

Many research efforts on self-adaptive systems assume that changes resulting in self-adaptation can be anticipated and the corresponding self-adaptation operations or behaviors can be determined and encapsulated into the software modules by developers. However, many self-adaptive systems are situated in open environments like the Internet or IoT (Bennaceur et al, 2016). Changes of the environment and the system itself cannot be totally anticipated. Engineering such complex systems is often difficult as the available knowledge at design time is not adequate to anticipate all the runtime conditions (Mahdavi-Hezavehi et al, 2016). Therefore, uncertainties of self-adaptive system should be taken as first-class concerns and the main challenges to develop modern complex self-adaptive systems. According to the viewpoint of Weyns, the fifth wave of evolution of self-adaptive system research is the "Guarantees Under Uncertainties". Esfahani analyzes various uncertainties in current self-adaptive systems (Weyns, 2017).

Undoubtedly, uncertainties of a self-adaptive system raise new open issues and challenge existing software engineering approaches. Decisions on self-adaptation operations should be made by software itself rather than developers, be postponed to run-time rather than design-time. To tackle the uncertainties has become an important trend in the field of software engineering of self-adaptive systems. There are several attempts and practices that intend to address the issues in term of some novel solutions, e.g., reinforcement learning (Amoui et al, 2008; Zhu et al, 2003). Learning methods are regarded as alternative and effective ways to address the uncertainty problems of self-adaptive system. Such approaches have been gained great attention and used in several aspects of engineering self-adaptive systems. Kim (Cernuzzi et al, 2005) proposes a reinforcement learning-based approach to online planning for robots in order to improve the robot's behavior by learning from prior experiences and by discovering adaptation plans in response to environmental changes. Fredericks (2016) proposes search-based techniques for hardening a self-adaptive software against uncertainty. Cámara et al. (2017) presents an analysis technique based on model checking of stochastic multi-player games that enables us to quantify the benefits in adaptation performance of factoring sensing uncertainty explicitly into decision-making.

A Sample of Self-Adaptive System

In order to understand the software requirements and the open issues of self-adaptive systems operating in open environment, this section first examines a sample of a hunting application.

The sample is designed to imitate the scenarios of hunting preys in open environment. The system consists of a number of agents that play various roles, e.g.: hunter and prey. They comprise a social organization. Each agent in the system can move and has limited resources (e.g., energy). The responsibility of a hunter agent is to catch prey agents as possible. When the energy of a hunter agent is under a specific threshold, it should move to a supply station to charge. The prey agent moves randomly in the dynamic and uncontrollable environment with various elements like sands, swamplands, and bedrock, etc. When moving in the environment, different quantities of energy may be consumed. The situation of the environment that agents operate may be uncertain and unpredictable. For example, unexpected obstacles may occur for some reasons, e.g., and dry land may be flooded, which becomes difficult to move and results in more energies to be consumed. Moreover, the consumed energies of agents also depend on the moving speed.

The hunter agents in the sample are expected with the capability to adapt to the environment changes in order to meet design objectives. For instance, to intercept a fast moving prey agent, hunter agent must adjust its movements accordingly, when energy is to be consumed, hunter agent must adjust it behaviors to find the place to recharge. In order to understand the diversity of self-adaptation and the design challenges, let us consider the following two scenarios.

- Self-adaptation by adjusting the behaviors.

In order to improve running efficiency and guarantee safety, hunter agent should take different behaviors according to the environment status and assigned tasks. For example, when moving in the sand road hunter agent should take the behavior of "highspeed-move" to decrease energy consumption and when moving in the marsh road hunter agent should take the behavior of "lowspeed-move" in case of overturning.

- Self-adaptation by adjusting the role.

In order to accomplish the assigned tasks effectively, hunter agent should take different roles depending on the status of the situated environment and itself. For example, when having enough energy and working well, hunter agent should take the role of "chaser" to find, chase and catch the preys. In such situation, hunter agent should perceive and interact with the preys in the environment. However, when hunter agent has not enough energy, it should play the role of "supplier" to get energy.

The above scenarios illustrate two kinds of self-adaptations accomplished at different granularity and level: coarse-grain self-adaptation by adjusting roles of an agent and fine-grain self-adaptation by adjusting behaviors of an agent.

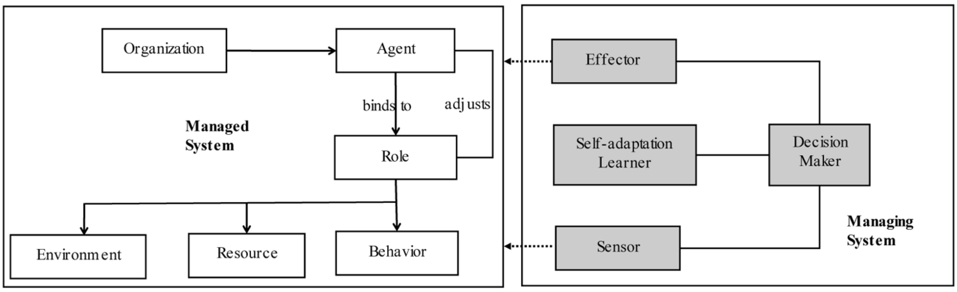
MODELS AND MECHANISMS OF SELF-ADAPTATION BASED ON REINFORCEMENT LEARNING

MAS Model of Self-Adaptive Systems

In our approach, the self-adaptive software is modeled as a multi-agent organization in which each software component is abstracted as an autonomous agent. Moreover, we use the organization metaphors to model and examine agent's behaviors and self-adaptation mechanisms because they give us high-level abstractions that enable us to specify and analyze self-adaptation while ignoring implementation details such as technologies and platforms, and, therefore, help to control the complexity of self-adaptive systems while simplifying software development.

A MAS is actually a social organization with a number of agents that play various roles of the organization and thereby take corresponding behaviors to accomplish the responsibilities assigned to the roles (see Figure 1). Each role defines the resources that it can provide, behaviors that it should take, and the environment that it is situated in. The behaviors of an agent in an organization depend on the roles the agent plays. However, an agent can adjust its behaviors encapsulated in the role according to the perceived environment in order to accomplish its assigned task in a better way. For example, in the above sample, the robot can take different behaviors to move (e.g., low-speed, medium-speed, and high-speed) according to

Figure 1. The model of self-adaptive multi-agent organization



the road condition, in order to save energy and guarantee safety. Such adjustment of behaviors represents a kind of self-adaptation at the agent and fine-grain level.

In MAS organization, an agent can play multiple roles and a role can be played by multiple agents. Playing a role means that an agent adopts the behavior, resource and environment features defined by the role. The assignment of roles can be achieved by a number of operations, including join, quit, activate and deactivate. When an agent joins a role, we say that the role is bound to by the agent. For example, the robot agent may play the role of "chaser" to catch the prey when it has full energy. Moreover, the bound role can be in different states, including active and inactive. When the role to which the agent binds is in an active state, it will govern the agent's behaviors, e.g., an agent takes behaviors based on the behavior specification of the role. Otherwise, when the role to which the agent binds is in the inactive state, it will not govern the agent's behaviors. However, an agent can access the resources of the role, or be sensitive to the environment defined in the role. An agent can take action quit to leaves the role. It also can change the status of the bound role by taking deactivate or activate actions. A formal definition and reasoning of dynamic binding can be found in (Mao et al, 2008). Another kind of self-adaptation is to adjust the behaviors of an agent according to the changes occurring in the situated environment or the agent. We call the above dynamic changes of the bound roles of agents as a dynamic binding mechanism, which represents a kind of self-adaptation at the organization and coarse-grain level.

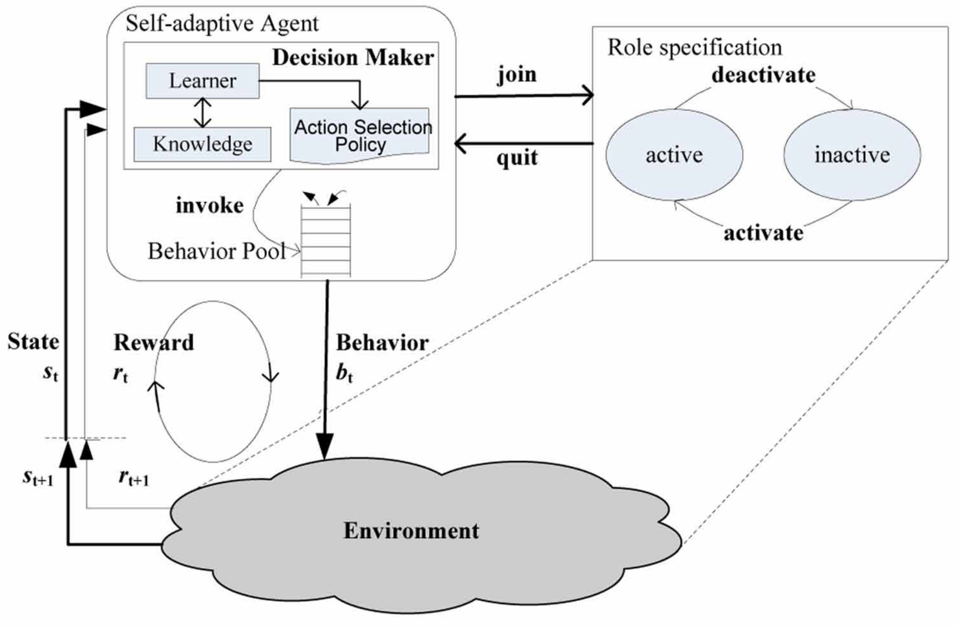
In summary, according to the model of self-adaptive software, the self-adaptations in MAS organization can be achieved in two manners: the coarse-grain and the fine-grain. In the coarse-grain manner, self-adaptation is completed by binding different roles in the organization and change their statue in order to adjust the position of agents in the organization. Obviously, when the roles of an agent are changed, its behaviors will be changed too. In the fine-grain manner, self-adaptation is completed by adjusting the accomplishment of the behaviors encapsulated in roles. Suck kind of self-adaptation will not change the position of the agent in the organization, i.e., the role played by the agent will not change.

Multiple-Layer Self-Adaptation Based on Reinforcement Learning

Designing self-adaptive software systems involves making design decisions about observing the environment and the system itself, selecting adaptation mechanisms, and enacting those mechanisms (Cheng et al, 2009). Typically, there are two types of decisions on self-adaptation: off-line and on-line. Off-line decision also called static decision is made by developers at design-time, which is widely adopted in most of self-adaptive approaches. In such approach, self-adaptation strategies or rules should be explicitly defined to specify the relationships between the expected environment changes and the self-adaptive adjustment of a system. Language facilities should be provided in order to describe the self-adaptation logic. Obviously, such approach is effective only when developers can clearly predict and explicitly define all changes of the environment and expected adjustments of the system at design-time. However, such requirements and assumptions typically cannot be satisfied for many self-adaptive systems that operate in an open environment. For such a kind of system, uncertainties issues occur due to the unpredictable event and uncontrollable evolving of the environment (Amoui et al, 2008). Learning-based on-line decision approach presents an alternative to address the problem. It enables a system itself to establish the relationship between the observed environment changes and the self-adaptive adjustment, and decide which adjustments should be made in term of learning at run-time.

We present an approach integrating self-adaptation mechanisms and reinforcement learning together to support on-line self-adaptation decision making for self-adaptive systems operating in open and unpredictable environment (see Figure 2). The agent in the system can not only take four meta-operations on roles to adjust its roles in the organization, but also can adjust its behaviors defined in the bound roles to respond to changes. The self-adaptation actions are transformed into a unified form

Figure 2. Self-adaptation decision based on reinforcement learning



of behavior, put into the behavior pool, and then scheduled by the behavior scheduler. The execution of behavior will influence the environment. The process of the agent adapting to the situated environment is similar to the interaction between an agent and its environment, in which the agent monitors the states of the environment and obtains rewards from the environment after it performs a behavior.

The agent selects a new behavior (b_t) based on the current state (s_t) and reward (r_t), after performing b_t, the environment transits into the next state (s_{t+1}) and produces a reward (r_{t+1}). This continuous interaction is the basis for self-adaptation. By interacting with the environment, an agent can learn the knowledge of the mappings between situations and adaptation actions, so as to dynamically and autonomously make decisions. A learner is used to implement the learning functionality according to a particular learning algorithm (e.g. Q algorithm); the "ε-greedy" selection strategy is used to choose an action from the action set. The learning result can be saved in various forms, one of which is a knowledge table that depicts the values of the actions on states.

DESIGN OF LEARNING ALGORITHMS FOR MULTIPLE-LAYER SELF-ADAPTATIONS

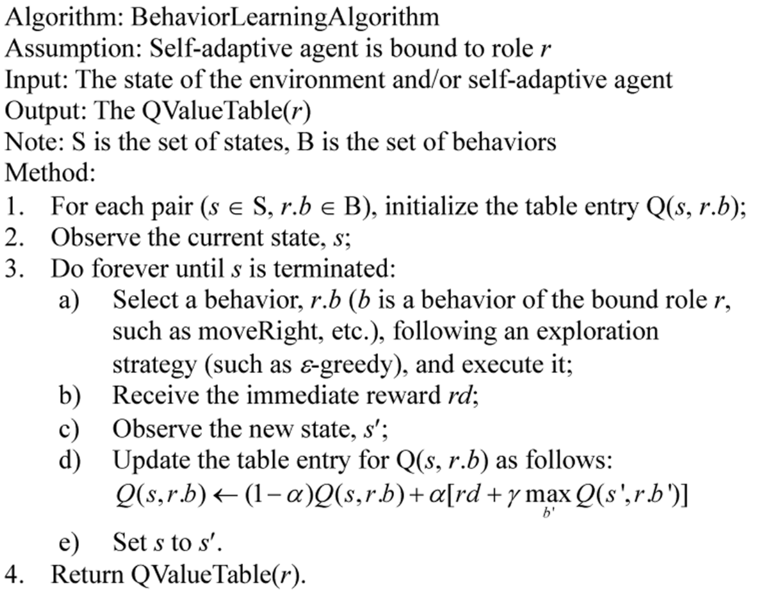
The purpose of learning on self-adaptation is to obtain the knowledge with which an agent can make on-line decisions to select appropriate self-adaptation operations in order to response to occurring changes that may be unanticipated at design-time. In this section, two self-adaptation learning algorithms for coarse-grain and fine-grain self-adaptations are designed respectively.

Learning Algorithm for Behavior Self-Adaptation

The adaptation learning algorithm for behaviors is designed based on the normal Q algorithm (see Figure 3). First, the concepts and elements of the normal Q algorithm are introduced and their specific meanings in our proposed algorithm are explained.

- **State:** The states in our algorithm include those of not only the environment, but also the self-adaptive agent itself.
- **Operation:** The optional operations relating to various states come from different roles (depicted as "*r.behavior*", "*r*" means a particular role and "*behavior*" means a behavior owned by *r*). The method to set the reward of a behavior is similar to that in the normal Q algorithm. The adaptation learning algorithm for behaviors is depicted in Figure 3, which has four steps. The input of the algorithm is the state of the environment and/or the self-adaptive

Figure 3. Adaptation learning algorithm for behaviors of roles



```
Algorithm: BehaviorLearningAlgorithm
Assumption: Self-adaptive agent is bound to role $r$
Input: The state of the environment and/or self-adaptive agent
Output: The QValueTable($r$)
Note: S is the set of states, B is the set of behaviors
Method:
1. For each pair ($s \in$ S, $r.b \in$ B), initialize the table entry Q($s$, $r.b$);
2. Observe the current state, $s$;
3. Do forever until $s$ is terminated:
    a) Select a behavior, $r.b$ ($b$ is a behavior of the bound role $r$,
       such as moveRight, etc.), following an exploration
       strategy (such as $\varepsilon$-greedy), and execute it;
    b) Receive the immediate reward $rd$;
    c) Observe the new state, $s'$;
    d) Update the table entry for Q($s$, $r.b$) as follows:
       $Q(s,r.b) \leftarrow (1-\alpha)Q(s,r.b)+\alpha[rd+\gamma \max_{b'} Q(s',r.b')]$
    e) Set $s$ to $s'$.
4. Return QValueTable($r$).
```

agent, and the output is a Q-table used to save the learning results when the agent binds to role *r*. The assumption made in the algorithm is that the agent has bound to role *r*.

First, a Q-table should be created and Q-values should be initialized. The Q-values are initialized as *0* in general, while the Q-values may be initialized as other meaningful values so that the learning progress can converge quickly. Second, the current state is observed and obtained, which is regarded as the initial state of the learning. Third, the learning loop should be performed until the state is terminated. Each loop contains the following steps like "*select a behavior — execute the behavior — receive a reward — observe a new state — update the Q-table — set the starting state of next cycle — select a behavior — … — update the Q-table — …*". Fourth, the Q-table (see Table 1) will be returned. In the table, the behavior set like {$r.b_1$, $r.b_2$,…, $r.b_n$}, the state set like {s_1, s_2, …, s_n}. Every item of Q(s_i, $r.b_j$) in the Q-table means the max reward for the execution of behavior b_j in state s_i.

Based on the Q-table, an agent can select appropriate operations in particular states. If the base rule of the policy is that the operation with the maximum Q-value is chosen, the Q-table can be mapped to the operation selection policy directly with the following formula:

Table 1. Q-table of Role r

	s_1	s_2	…	s_n
r.b_1	Q(s_1, r.b_1)	Q(s_2, r.b_1)	…	Q(sn, r.b_1)
r.b_2	Q(s_1, r.b_2)	Q(s_2, r.b_2)	…	Q(sn, r.b_2)
…	…	…	…	…
r.b_m	Q(s_1, r.b_m)	Q(s_2, r.b_m)	…	Q(sn, r.b_m)

$$\pi(s_i) = a_k, a_k \in \{a \mid Q(s_i, a) = \max_j Q(s_i, a_j)\}$$

As the formula shows, the max Q-value operation α_k which is $r.b_k$ in the Q-table actually is chosen according to the state s_i.

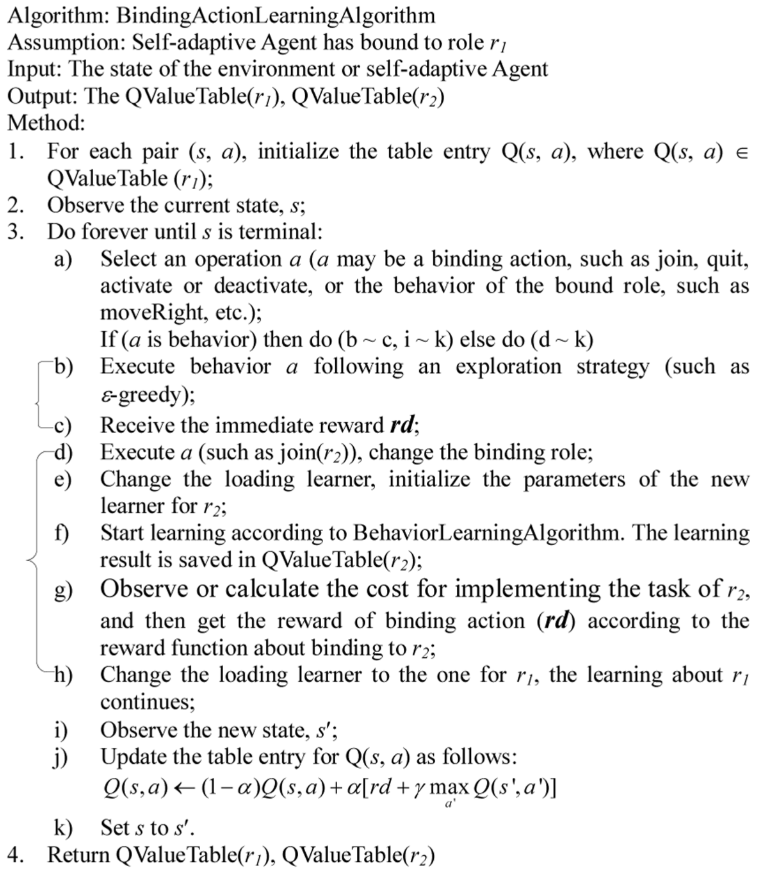
Learning Algorithm for Role Self-Adaptation

As for the coarse-grain self-adaptation at the organization level, the decisions on self-adaptation is to choose appropriate operations to dynamically bind to roles of the MAS organization (see Figure 4). The following describes specific concepts used in this algorithm.

- **State:** like the one described in section MAS model of self-adaptive systems.
- **Operation:** In the normal Q algorithm, the optional operations related to various states have only one granularity which is like the fine-grain operations in our method. However, the operations in our algorithm include fine-grain and coarse-grain ones. The fine-grain operations include the behaviors owned by various roles, and the coarse-grain operations include "*join*", "*quit*", "*activate*", "*deactivate*" meta-operations on a role.
- **Reward Function:** Due to the different granularities of the operations, the reward function corresponding to these operations should be set in different ways. For the reward function corresponding to behavior, the setting method is similar to the one in the normal Q algorithm. However, because of the property of delayed reward of coarse-grain operations, the setting method of the reward function is different from the fine-grain ones. In the followings, we take the "*join*" operation as an example to illustrate the setting method.

The reward of an action is set according to the degree of the effect on the environment caused by the execution of the action in a particular state. However, the execution of the "*join*" operation cannot affect the environment directly. And it

Figure 4. Adaptation learning algorithm for the operations on bound roles



```
Algorithm: BindingActionLearningAlgorithm
Assumption: Self-adaptive Agent has bound to role $r_1$
Input: The state of the environment or self-adaptive Agent
Output: The QValueTable($r_1$), QValueTable($r_2$)
Method:
1. For each pair ($s$, $a$), initialize the table entry Q($s$, $a$), where Q($s$, $a$) $\in$
   QValueTable ($r_1$);
2. Observe the current state, $s$;
3. Do forever until $s$ is terminal:
    a) Select an operation $a$ ($a$ may be a binding action, such as join, quit,
       activate or deactivate, or the behavior of the bound role, such as
       moveRight, etc.);
       If ($a$ is behavior) then do (b ~ c, i ~ k) else do (d ~ k)
    b) Execute behavior $a$ following an exploration strategy (such as
       $\varepsilon$-greedy);
    c) Receive the immediate reward **rd**;
    d) Execute $a$ (such as join($r_2$)), change the binding role;
    e) Change the loading learner, initialize the parameters of the new
       learner for $r_2$;
    f) Start learning according to BehaviorLearningAlgorithm. The learning
       result is saved in QValueTable($r_2$);
    g) Observe or calculate the cost for implementing the task of $r_2$,
       and then get the reward of binding action (**rd**) according to the
       reward function about binding to $r_2$;
    h) Change the loading learner to the one for $r_1$, the learning about $r_1$
       continues;
    i) Observe the new state, $s'$;
    j) Update the table entry for Q($s$, $a$) as follows:
       $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[rd + \gamma \max_{a'} Q(s',a')]$
    k) Set $s$ to $s'$.
4. Return QValueTable($r_1$), QValueTable($r_2$)
```

affects the environment when the behaviors owned by the role have been executed. Hence, the execution of the "*join*" operation will cause the delayed reward problem, which is not the same as the one in the normal Q algorithm.

The adaptation learning algorithm for the operations on binding roles is shown in Figure 4. The input of this algorithm is the state of the environment and/or the self-adaptive agent, and the outputs are two Q-tables used to save the learning results when agent binds to different roles. The assumption made in the algorithm is that the agent is bound to the role r_1.

First, A Q-table should be created and Q-values should be initialized. The Q-values are initialized as 0 in general, while the Q-values may be initialized as other meaningful values so that the learning progress can converge quickly. Second, the current state should be observed and obtained, which is regarded as the initial state of the learning. Third, the learning loop should be performed until the state is terminated.

The learning algorithm about behaviors (in steps *a~c* and *i~k*) and binding to roles (in steps a and *d~k*) is included in this loop. The execution of the chosen self-adaptation actions may result in the self-adaptive agent to adjust its bound roles. For example, the bound role is changed from r_1 to r_2. At the same time, different learners and Q-tables related to r_2 will be loaded. After that, the agent should learn the execution of the behavior of r_2 according to the learning algorithm about the behavior. When the execution of the behaviors in r_2 terminates, the whole cost (e.g. moving steps or consuming energy) which is used to achieve the objective or accomplish the task will be calculated. Based on the cost and the reward function, the reward of bound role r_2 can be calculated. After performing the behavior of r_2 or being unable to execute the behaviors of r_2 due to some reasons, the agent will change its role to r_1 so that the learning process about r_1 continues.

At the learning phase, agent accumulates the experiences by updating $Q(s_t,a_t)$, and the update process can be depicted with the following formula.

$$Q\left(s,\alpha\right) \leftarrow \left(1-\alpha\right)Q\left(s,\alpha\right)+\alpha\left[rd+\gamma \max_{\alpha'} Q\left(s',\alpha'\right)\right]$$

In the formula, the term $\left(1-\alpha\right)Q\left(s,\alpha\right)$ represents the value the agent already knows and its weight for accumulation, where α controls the weight. On the other hand, the term $\alpha\left[rd+\gamma \max_{\alpha^1} Q\left(s',\alpha'\right)\right]$ represents the current reward and the expectation of the greedy action of the newly observed state where α $(0 \leq \alpha \leq 1)$ controls the term's weight and γ $(0 < \gamma \leq 1)$ controls the weight of the greedy action.

Last, Two Q-tables will be returned (seeing Table 2). The agent can select appropriate operations in different states according to the Q-table similar to section MAS model of self-adaptive systems.

Table 2. Q-table of Role r1

	s_1	s_2	…	s_n
$r_1.b_1$	$Q(s_{1,}\ r_1.b_1)$	$Q(s_2, r_1.b_1)$	…	$Q(s_n, r_1.b_1)$
…	…	…	…	…
$r_1.b_m$	$Q(s_1, r_1.b_m)$	$Q(s_2, r_1.b_m)$	…	$Q(s_n, r_1.b_m)$
$join(r_2)$	$Q(s_1, join(r_2))$	$Q(s_2, join(r_2))$	…	$Q(s_n, join(r_2))$

SOFTWARE ARCHITECTURE AND DEVELOPMENT PROCESS OF SELF-ADAPTIVE MAS

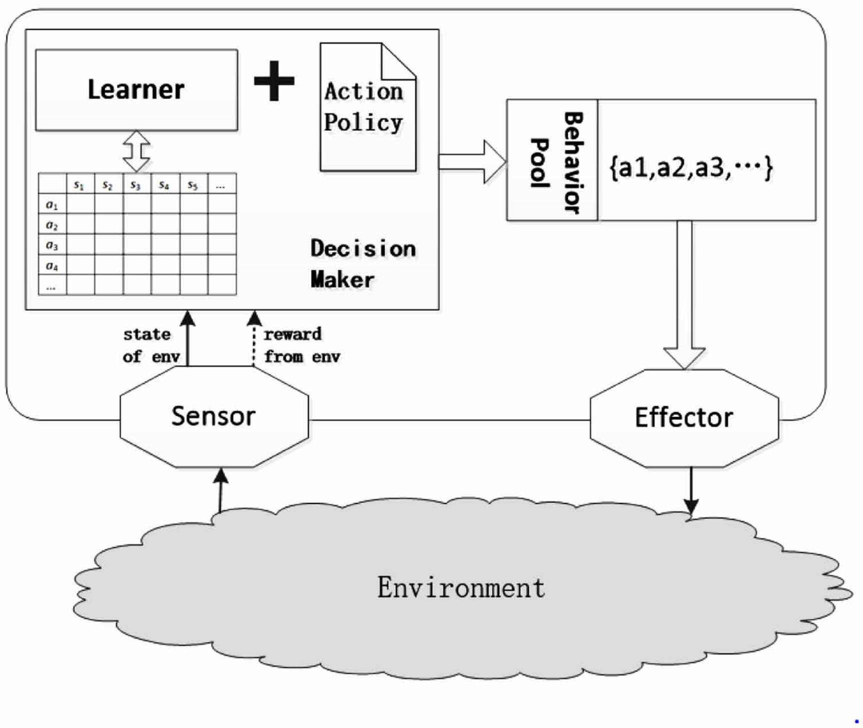
This section introduces the development technologies for self-adaptive software with a development study, including the software architecture and development process.

Software Architecture of Self-Adaptive Agents

In order to implement the elementary software components in self-adaptive systems with the models and mechanisms described in the above sections, we propose a software architecture for self-adaptive agents that have the on-line self-adaptation capability based on a self-adaptive mechanism and reinforcement learning (see Figure 5). A self-adaptive agent is composed of a number of software units that can be in the form as a sensor, a learner, a knowledge table, a behavior pool, and an effectors.

A self-adaptive agent can perceive the environment changes in term of sensors. The perceived information is sent to the decision makers to determine which self-adaptation actions should be taken according to the perceived states and the

Figure 5. Software architecture of self-adaptive agent

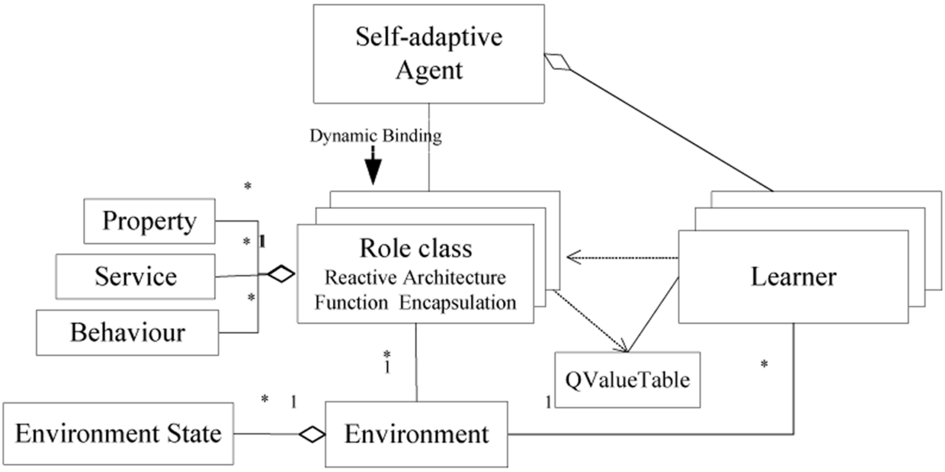


rewards from the environment. The selected self-adaptation actions are put into the behavior pool so as to be executed by the behavior scheduler. After the behaviors are executed, the environment states may be changed too. This change triggers a new self-adaptation closed loop.

In order to support the development of self-adaptive multi-agent systems, a software framework called SADE is implemented. The framework provides a method and reusable software package to guide the construction of complex self-adaptive systems. It encapsulates a number of fundamental functions that accomplish the multiple-layer self-adaptation and reinforcement learning-based self-adaptation decision making. Typically, a program of self-adaptive software adopting our approach can be implemented into four parts (see Figure 6).

- **Self-Adaptive Agent (SAgent):** SAgent is an abstract class that should be inherited when constructing adaptive agents. The reusable component provides basic self-adaptation capability at both coarse-grain and fine-grain levels. It encapsulates a number of meta-operations on self-adaptation and some fundamental functionalities, such as lifecycle management for self-adaptation, an event queue and a scheduler.
- **Role:** Role is the elementary module of self-adaptive software. It organizes and encapsulates the functions and behaviors of a self-adaptive system at application-level. Developers should encapsulate high-coupled behaviors, events, properties and services together as roles by inheriting the base class "Role".

Figure 6. The development framework to construct self-adaptive MAS



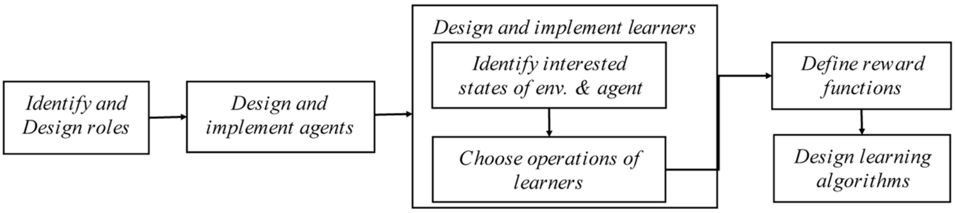
- **Learner:** Learner is a base class that should be inherited by various learners. Each SAgent owns an instance of the Learner class. At run-time, a self-adaptive agent can load a specific learner by changing the type of the instance to the specific learner class. The learner class is used by a self-adaptive agent to learn how to adapt to the environment changes. The learning results are saved as a knowledge table (e.g., QValueTable). To accomplish different objectives, a self-adaptive agent may learn the knowledge about the environment by loading different learners. In our approach, the role behaviors of application logic and the learning behaviors of learning logics are encapsulated in different modules. We believe that such a separation is helpful for programmers to focus on the different aspects of the program when developing complex self-adaptive software, and make the self-adaptive software update easily, and therefore simplifies the development and maintenance.
- **Environment:** The environment module is an abstract or simulation one. It has states which may be changed dynamically. When the SAgent takes actions by the effector, the states of the environment might change and the rewards will be returned to the SAgent.

Development Process and Case Study

In order to support the systematic development of complex self-adaptive software, we propose a development process, with which we study a case described in section RELATED WORKS to illustrate how to develop self-adaptive systems with our approach. To simplify the design and implementation, some assumptions about the sample are made. First, the robot agents and prey agents are situated and move around in a 10×10 grid. Second, obstacles in the environment are generated automatically and unpredictably. Third, in each step, an agent has four possible actions to choose from {moving up, moving down, moving left, moving right}, and the hunter agents can sense the moving directions of the prey agents.

The software development process for a self-adaptive system consists of the following steps (see Figure 7).

Figure 7. Development process of self-adaptive software


Identify and Design roles
Design and implement agents
Design and implement learners
Identify interested states of env. & agent
Choose operations of learners
Define reward functions
Design learning algorithms

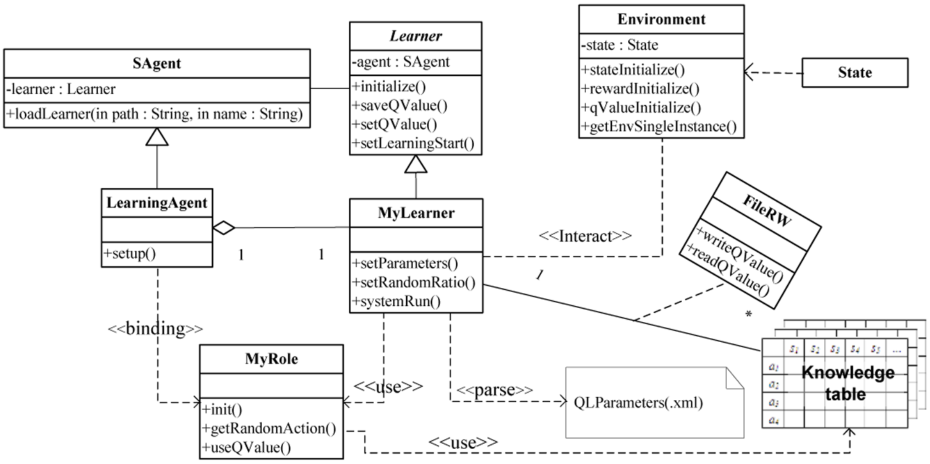
Step 1: Identify and Design Roles

Developers should identify and design the roles of the system by inheriting abstract class *Role*. The properties, behaviors and internal events should be explicitly defined. In the sample, two roles of "Catcher" and "Supplier" can be identified. The catcher role encapsulates a number of behaviors and properties related with a catcher, such as "*moving*", "*sensing a prey agent in the environment*", etc. The supplier role encapsulates some behaviors to search the supply station, and charge the agent. Figure 8 depicts the class diagram of the Supplier role.

Step 2: Design and Implement Agents

Developers should design and implement agents in multi-agent systems. If the agent has the capability of self-adaptation, it should inherit abstract class *SAgent*. In the sample, there are two kinds of agents: *HunterAgent* and *PreyAgent*. *HunterAgent* is a self-adaptive agent and is designed to catch preys. The program of *HunterAgent* should explicitly declare the class path of the roles that the agent will join and the learner that the agent will load. When a *HunterAgent* is created, it can obtain the capabilities and behaviors to catch a prey agent by joining the Catcher role.

Figure 8. Part of the design class diagrams of the sample



Step 3: Design and Implement Learners

A developer should design and implement the learners that support on-line self-adaptation decisions. Two learners of "*CatchLearner*" and "*SupplyLearner*" can be identified and designed in the sample. The *CatchLearner* is responsible for learning the knowledge of catching a prey agent, such as how to move when encountering unanticipated obstacles or sensing the moving direction changes of the prey agents. The *SupplyLearner* is responsible for learning how to go to the supply station for charging with the least energy consumption. The step of designing learners can be divided into four sub-steps.

Step 3.1: Identify Interested States of Environment or Agent

The number of the states may be very large for many complex systems. However, only some of them may influence the learning process, while others may not. Therefore, it is very important for developers to select appropriate and enough states which are meaningful and effective for learning. In this case, the interested states for learning include the position of the hunter agent, the moving direction of the prey agent, the information about obstacles around the hunter agent, and the energy level of it.

Step 3.2: Choose Operations of the Learner

A learner learns the knowledge about which action should be chosen in a specified situation. In our approach, the actions that the learner can choose include two types: the coarse-grain self-adaptation actions and the fine-grain ones. Hence, the *CatchLearner* can choose the actions for adjusting roles (e.g., *join*, *quit*) and the behaviors defined in roles such as moving left, right, up and down.

Step 3.3: Define Reward Function

It is very important for a learner to set a proper reward value for each action. In this case, we design the reward rules for the hunter agent as follows. For example, it will get -1 when it moves one step without catching the prey agent; it will get different negative values when it hits on or passes different obstacles. It will get -2 when it moves one step in sand and -3 when it moves one step in mud; and it will get 100 when it catches the prey agent. The reward of binding a role is defined as follows.

$$\text{reward} = \begin{cases} 50 \,/\, remainEnergy,\ \mathit{if}\, remainEnergy \leq 50. \\ -100,\ \mathit{if}\, remainEnergy > 50\, \mathit{or\,moving\,to\,supplystationunsuccessful}. \end{cases}$$

Step 3.4: Design Learning Algorithm

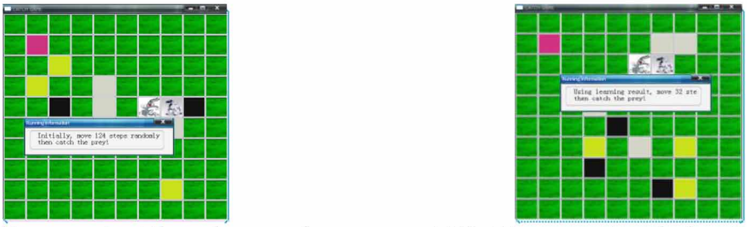
The Q-learning algorithm adopted in this case is as shown in section DESIGN OF LEARNING ALGORITHMS FOR MULTIPLE-LAYER SELF-ADAPTATIONS. *LearningAgent* owns an instance of Learner, and it can change the type of the instance to *SupplyLearner* by downcast. *SupplyLearner* is responsible for learning the knowledge about how to go to supply station with the least energy consumption. The learning results are saved in a knowledge table which can be used by the *Supplier* to guide its moving actions.

We have successfully developed the software prototype (see Figure 9) for the sample with our SADE (Self-adaptive Agent Development Environment) (Dong et al, 2009) platform that is dedicated to developing self-adaptive multi-agent systems based on the approaches presented in this paper. Figure 9 (a) shows the hunter agent moves 124 steps before it catches a prey agent at the early stage of learning. After the hunter agent tries many times to learn the knowledge about the dynamic and uncertain environment, it catches a prey agent within only 33 steps (shown in Figure 9 (b)).

EXPERIMENT EVALUATION

This section introduces experiments to demonstrate the effectiveness of the proposed approach. Our experiments are conducted in two catching scenarios. The first is to show how a hunter agent catches more preys with the least moving steps in a dynamic environment (grid) in which the obstacles and positions of preys are changing unpredictably when it binds to the catcher role. The second is about how a hunter agent can catch more preys with limited energy by changing its role. In the sample, the energy of a hunter agent will decrease continually when it moves in the grid. To supply energy in time so as to continue the catching activity, the hunter

Figure 9. Snapshot of the case study



```
Initially, move 124 steps randomly
then catch the prey!
```

```
Using learning result, move 32 ste
then catch the prey!
```

(a) Catching scenario at the early stage of learning

(b) Catching scenario at the late stage of learning

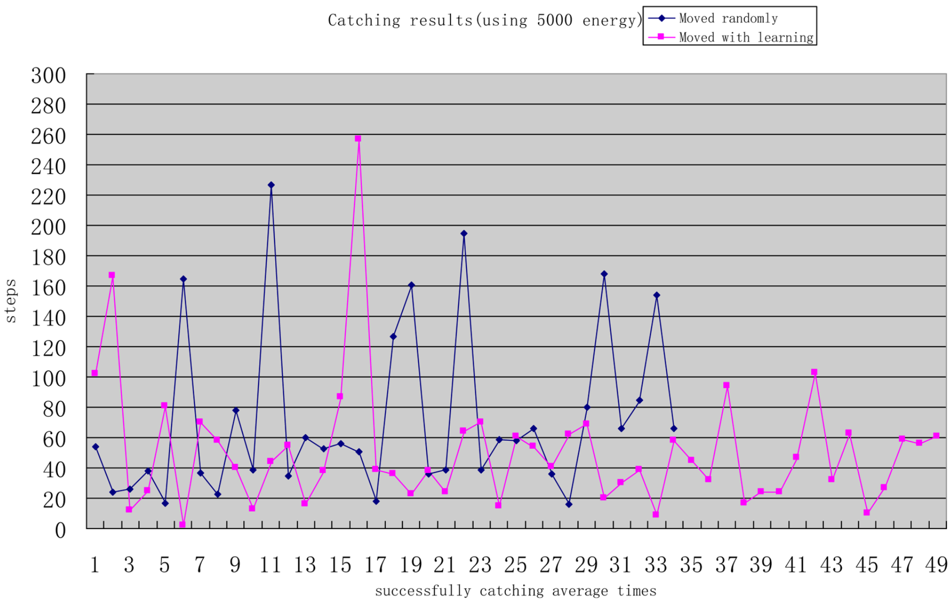
agent must change its role to search for the supply station. In various situations, the hunter agent will need a different amount of energy to arrive at the supply station. Therefore, it is important to give the appropriate Q-values of the pairs of the states and changing role operations so that the hunter agent can change its role to supplier in time and can arrive at the supply station successfully.

Scenario 1: One hunter agent catches preys when it binds to the catcher role within enough energy. In this scenario, a hunter agent catches a prey agent randomly or by learning.

Experiment 1.1: Preys are running on the grid. The parameters related to the learning are set as follows: the energy owned by the hunter agent is 5000 units, $\alpha=1$, $\gamma=0.9$, $\varepsilon=0.8$.

Figure 10 shows the result of the hunter agent catching prey agents in two different ways: moving randomly and moving with learning, when the hunter agent joins the catcher role and owns enough energy. The numbers at the horizontal axis mean the average times the hunter agent successfully catches a prey agent. The numbers in the longitudinal axis mean the steps the hunter agent moves before it catches a prey agent. In Figure 10, we can find that a hunter agent can catch a prey agent with fewer steps mostly by learning than moving randomly. However, because the action

Figure 10. Catching prey results in two ways in the first experiment

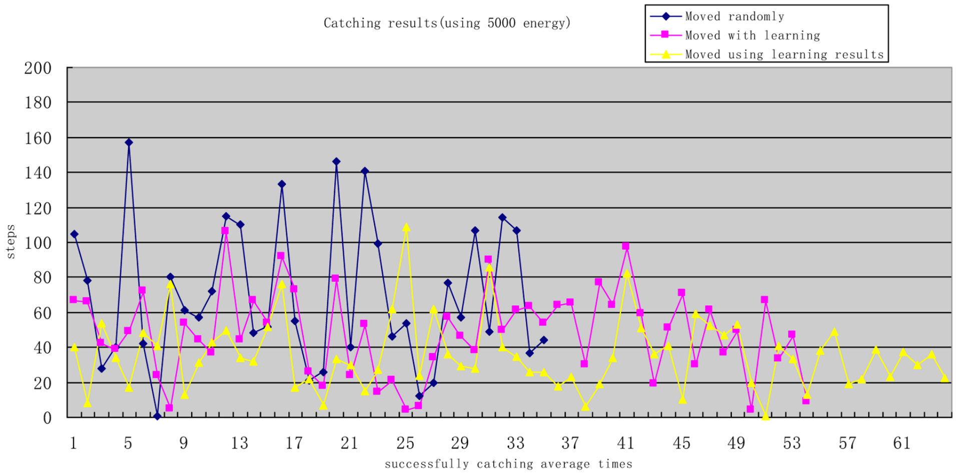


selection policy is the "ε-greedy" selection strategy where ε=0.8, the hunter agent explores the new position with 0.2 probability, which makes the steps the hunter agent moves to catch a prey agent with learning may be more than the steps the hunter agent moves randomly, especially at the early phase of learning. The phenomenon occurs in the following experiments, while with more and more learning times, the effect of catching a prey agent with learning is better than that of moving randomly.

Experiment 1.2: Figure 11 depicts the catching results in three ways based on Experiment 1.1. Different from Figure 8, one yellow curved line is added to describe the situation of the hunter agent catching the prey agent by learning (ε=1). In Figure 11, we can see that the catching result by selecting the maximum Q-value operation in every state may become better after learning many times.

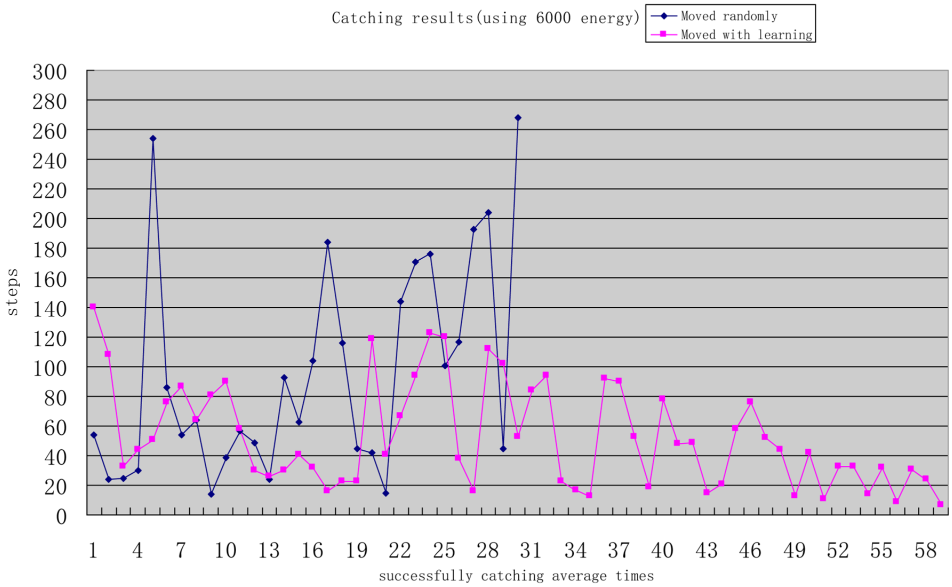
Experiment 2.1: The prey agent changes its moving from around the grid to along a path to do circular motion over time. The energy owned by the hunter agent is 6000 units, and other parameters in this experiment are same as the ones described in Experiment 1.1. In Figure 12, we can find that the consuming steps are more after 16 at the horizontal axis. After many times of learning, the number of steps used to catch prey agent is reduced gradually. The reason why the consuming steps increase sharply is that the prey agent changes its running, the previous knowledge about the environment is old and not completely correct. However, the agent can update the knowledge continually by interacting with the environment so that it can change the adaptation policy on-line.

Figure 11. Catching prey results in three ways based on the first experiment



**For a more accurate representation see the electronic version.*

Figure 12. Catching prey results in two ways in the third experiment

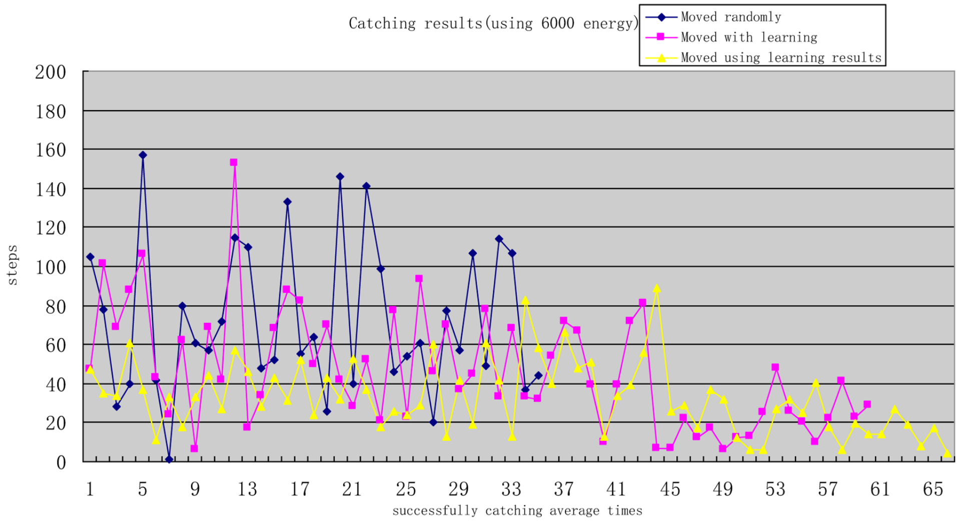


Experiment 2.2: Similar to Figure 11, Figure 13 shows the catching results in three ways based on experiment 2.1. The yellow curved line describes the situation of the hunter agent catching the prey agent by learning (ε=1), which shows a better result than the other two ways.

Scenario 2: To guarantee catching more prey agents and supplying energy in time, a hunter agent should learn the best opportunity for changing role. In this scenario, the hunter agent can change its role in two different ways. One way is that the hunter agent changes its role to supplier when its amount of energy is less than the fixed value (e.g. 40 units). The other way is that the hunter agent changes its role at a different energy level, which should be learned.

Experiment 2.1: The parameters related to the learning are set as follows: the energy owned by hunter agent is 200 units, α=1, γ=0.9, ε=0.8. In this experiment, we count two elements in every execution: whether the agent supplies energy successfully or not and the number of captured preys. We compare the results of the 5th, 10th and 15th 100 times of execution in the followings. Figure 14 shows the comparisons of the times of successfully supplying energy in two ways. Obviously, they are becoming more as the learning continues and the times in the second way are more than the times in the first way. Figure 15 shows the average numbers of captured preys in different ways. We can find that the average numbers in the second way are larger than those in the first

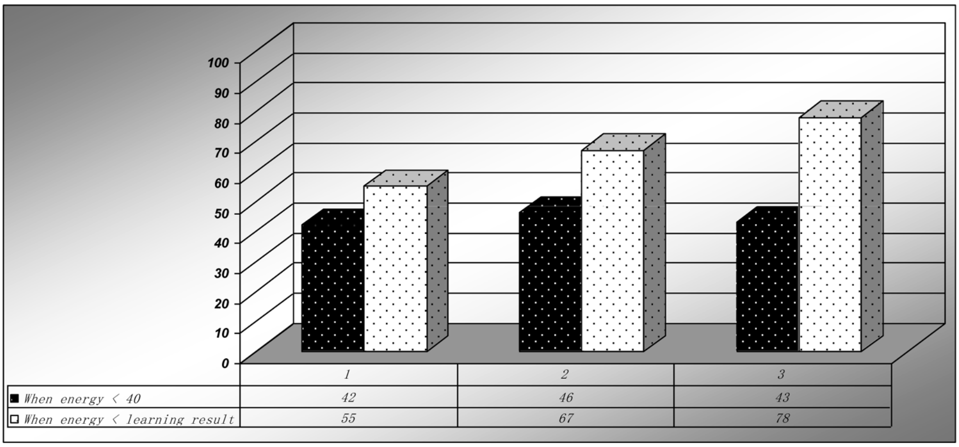
Figure 13. Catching prey results in three ways based on the third experiment



**For a more accurate representation see the electronic version.*

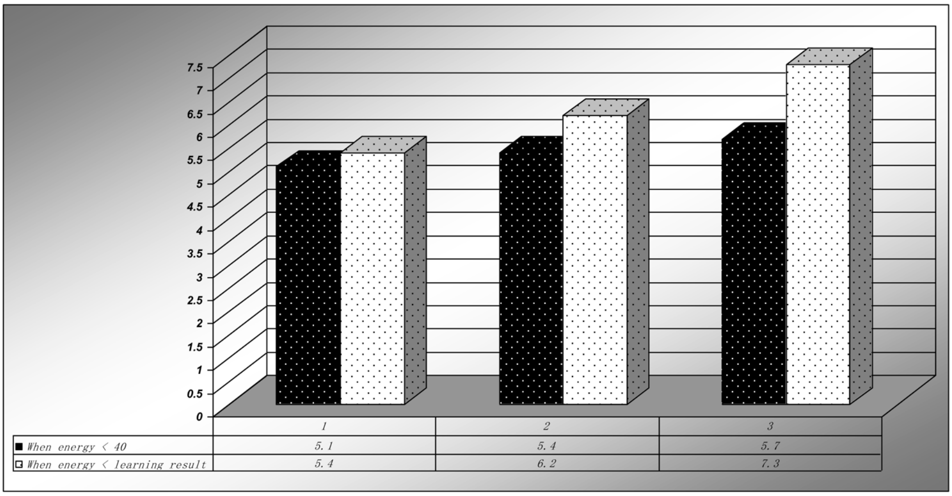
way because the successfully supplying energy times in the second way are more than the times in first way so that the average energy used to catch a prey agent in the second way is more than that in the first way. This experiment shows that the hunter agent can sense its energy level, and it will change its role in time when it judges that the energy is in critical.

Figure 14. Times comparisons of supplying energy successfully when the agent changes its role in different ways



	1	2	3
■ When energy < 40	42	46	43
□ When energy < learning result	55	67	78

Figure 15. Average count comparisons of catching prey in different ways



	1	2	3
■ When energy < 40	5.1	5.4	5.7
□ When energy < learning result	5.4	6.2	7.3

CONCLUSION

The self-adaptive system operating in open environment may be extremely complex. The complexity stems from several aspects, including (1) the diversity of self-adaptation occurring at different levels and granularities; (2) the uncertainty of self-adaptation resulting from the unpredictable and uncontrollable environments. To develop such complex self-adaptive system needs high-level abstractions, effective mechanisms and engineering approaches.

This paper attempts to address the software engineering challenges arising from the self-adaptive system with the diversity and uncertainty features. We investigate the fine-grain and coarse-grain self-adaptions and seek effective abstractions, models, and mechanism to implement such kind of self-adaptive system.

1. An abstract model of self-adaptive system. Based on the metaphors of MAS organizations, we present an abstract model of self-adaptive to examine the multiple-layer self-adaptations and their properties. The coarse-grain self-adaptation is accomplished in term of the dynamic binding of role at organization level. Four meta-operations on self-adaptation are proposed, including join, quit, activate and deactivate. The fine-coarse self-adaptation is accomplished in term of the adjustment of the behaviors defined in the role.
2. Self-adaptation decision based on reinforcement learning. In order to tackle the issues of uncertainties at different self-adaptation layer, we propose on-line decision methods for different layer self-adaptations based on reinforcement

learning. Such method enables software agents to make decisions on self-adaptation behaviors by themselves at run-time. The learning algorithms for decisions are designed and the decision processes are detailed.

3. Development supports for self-adaptive software. In order to support the engineering of complex self-adaptive software, we have implemented a software framework called SADE that provides a reusable software package for implementing self-adaptive software. The software architecture integrates the organization metaphors and learner. A software development process based on the SADE is proposed to guide the developers in a systematics way.

We have developed a sample of self-adaptive systems based on the above technique in detail, and conduct several experiments to validate the effectiveness and efficiency of the whole technical framework proposed in the paper.

REFERENCES

Abbas, N., Andersson, J., Iftikhar, M. U., & Weyns, D. (2016, April). Rigorous architectural reasoning for self-adaptive software systems. In Software Architectures (QRASA), 2016 Qualitative Reasoning about (pp. 11-18). IEEE. doi:10.1109/QRASA.2016.9

Amoui, M., Salehie, M., Mirarab, S., & Tahvildari, L. (2008, March). Adaptive action selection in autonomic software using reinforcement learning. In *Autonomic and Autonomous Systems, 2008. ICAS 2008. Fourth International Conference on* (pp. 175-181). IEEE. 10.1109/ICAS.2008.35

Bennaceur, A., McCormick, C., Galán, J. G., Perera, C., Smith, A., Zisman, A., & Nuseibeh, B. (2016, May). Feed me, feed me: An exemplar for engineering adaptive software. In *Proceedings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems* (pp. 89-95). ACM. 10.1145/2897053.2897071

Cámara, J., Correia, P., de Lemos, R., Garlan, D., Gomes, P., Schmerl, B., & Ventura, R. (2016). Incorporating architecture-based self-adaptation into an adaptive industrial software system. *Journal of Systems and Software*, *122*, 507–523. doi:10.1016/j.jss.2015.09.021

Cámara, J., Peng, W., Garlan, D., & Schmerl, B. (2017) Reasoning about Sensing Uncertainty in Decision-Making for Self-Adaptation. *Proceedings of the 15th International Workshop on Foundations of Coordination Languages and Self-Adaptive Systems*.

Cernuzzi, L., & Zambonelli, F. (2005, July). Dealing with adaptive multi-agent organizations in the gaia methodology. In *International Workshop on Agent-Oriented Software Engineering* (pp. 109-123). Springer.

Cernuzzi, L., & Zambonelli, F. (2011). Adaptive organizational changes in agent-oriented methodologies. *The Knowledge Engineering Review*, *26*(2), 175–190. doi:10.1017/S0269888911000014

Cheng, B. H., De Lemos, R., Giese, H., Inverardi, P., Magee, J., Andersson, J., & Serugendo, G. D. M. (2009). Software engineering for self-adaptive systems: A research roadmap. In *Software engineering for self-adaptive systems* (pp. 1–26). Springer Berlin Heidelberg. doi:10.1007/978-3-642-02161-9_1

Dastani, M., Van Riemsdijk, M. B., Hulstijn, J., Dignum, F., & Meyer, J. J. C. (2004, July). Enacting and deacting roles in agent programming. In *International Workshop on Agent-Oriented Software Engineering* (pp. 189-204). Springer.

Dong, M., Mao, X., Yin, J., Chang, Z., & Qi, Z. (2009). SADE: a development environment for adaptive multi-agent systems. *Principles of Practice in Multi-Agent Systems*, 516-524.

Esfahani, N., & Malek, S. (2013). Uncertainty in self-adaptive software systems. In *Software Engineering for Self-Adaptive Systems II* (pp. 214–238). Springer Berlin Heidelberg. doi:10.1007/978-3-642-35813-5_9

Filieri, A., Maggio, M., Angelopoulos, K., D'ippolito, N., Gerostathopoulos, I., Hempel, A. B., ... Krikava, F. (2017). Control Strategies for Self-Adaptive Software Systems. *ACM Transactions on Autonomous and Adaptive Systems*, *11*(4), 24. doi:10.1145/3024188

Fredericks, E. M. (2016, May). Automatically hardening a self-adaptive system against uncertainty. In *Proceedings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems* (pp. 16-27). ACM. 10.1145/2897053.2897059

Garlan, D., Cheng, S. W., Huang, A. C., Schmerl, B., & Steenkiste, P. (2004). Rainbow: Architecture-based self-adaptation with reusable infrastructure. *Computer*, *37*(10), 46–54. doi:10.1109/MC.2004.175

Gerostathopoulos, I., Bures, T., Hnetynka, P., Keznikl, J., Kit, M., Plasil, F., & Plouzeau, N. (2016). Self-adaptation in software-intensive cyber–physical systems: From system goals to architecture configurations. *Journal of Systems and Software*, *122*, 378–397. doi:10.1016/j.jss.2016.02.028

Gil de la Iglesia, D., & Weyns, D. (2013, May). SA-MAS: Self-adaptation to enhance software qualities in multi-agent systems. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems* (pp. 1159-1160). International Foundation for Autonomous Agents and Multiagent Systems.

Hilaire, V., Koukam, A., & Gruer, P. (2002, October). A mechanism for dynamic role playing. In *Net. ObjectDays: International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World* (pp. 36-48). Springer.

Jensen, A. S., & Villadsen, J. (2013). A comparison of organization-centered and agent-centered multi-agent systems. *Artificial Intelligence Review*, *2*(3), 59.

Kim, D., & Park, S. (2009, May). Reinforcement learning-based dynamic adaptation planning method for architecture-based self-managed software. In *Software Engineering for Adaptive and Self-Managing Systems* (pp. 76–85). IEEE.

Lee, S., Oh, J., & Lee, E. (2005, July). *An Architecture for Multi-agent Based Self-adaptive System in Mobile Environment*. IDEAL. doi:10.1007/11508069_64

Litoiu, M., Shaw, M., Tamura, G., Villegas, N. M., Müller, H., Giese, H., & Rutten, E. (2017). What Can Control Theory Teach Us About Assurances in Self-Adaptive Software Systems? *Software Engineering for Self-Adaptive Systems 3: Assurances*, 9640.

Mahdavi-Hezavehi, S., Avgeriou, P., & Weyns, D. (2016). A classification framework of uncertainty in architecture-based Self-adaptive systems with multiple quality requirements. *Managing Trade-offs in Adaptable Software Architectures*, 45-78.

Mahdavi-Hezavehi, S., Durelli, V. H., Weyns, D., & Avgeriou, P. (2017). A systematic literature review on methods that handle multiple quality attributes in architecture-based self-adaptive systems. *Information and Software Technology*, *90*, 1-26.

Mao, X., Shan, L., Zhu, H., & Wang, J. (2008). An adaptive casteship mechanism for developing multi-agent systems. *International Journal of Computer Applications in Technology*, *31*(1-2), 17–34. doi:10.1504/IJCAT.2008.017716

Muccini, H., Sharaf, M., & Weyns, D. (2016). Self-adaptation for cyber-physical systems: a systematic literature review. In *Proceedings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems* (pp. 75-81). ACM. 10.1145/2897053.2897069

Odell, J. J., Parunak, H. V. D., Brueckner, S., & Sauter, J. (2003, July). Temporal aspects of dynamic role assignment. In *International Workshop on Agent-Oriented Software Engineering* (pp. 201-213). Springer.

Pandey, A., Moreno, G. A., Cámara, J., & Garlan, D. (2016, September). Hybrid planning for decision making in self-adaptive systems. In *Self-Adaptive and Self-Organizing Systems (SASO), 2016 IEEE 10th International Conference on* (pp. 130-139). IEEE. 10.1109/SASO.2016.19

Popescu, R., Staikopoulos, A., Brogi, A., Liu, P., & Clarke, S. (2012). A formalized, taxonomy-driven approach to cross-layer application adaptation. *ACM Transactions on Autonomous and Adaptive Systems*, *7*(1), 7. doi:10.1145/2168260.2168267

Sabatucci, L., Seidita, V., & Cossentino, M. (2017, June). The Four Types of Self-adaptive Systems: A Metamodel. In *International Conference on Intelligent Interactive Multimedia Systems and Services* (pp. 440-450). Springer.

Salehie, M., & Tahvildari, L. (2009). Self-adaptive software: Landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems, 4*(2), 14.

Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., & Paige, R. (2012). Large-scale complex IT systems. *Communications of the ACM*, *55*(7), 71–77. doi:10.1145/2209249.2209268

Viana, M., Alencar, P., & Lucena, C. (2016). A Modeling Language for Adaptive Normative Agents. In *Multi-Agent Systems and Agreement Technologies* (pp. 40–48). Cham: Springer.

Weyns, D. (2017). Software engineering of Self-adaptive systems: an organised tour and future challenges. In Handbook of Software Engineering. Springer.

Zhu, H., & Lightfoot, D. (2003, August). Caste: A step beyond object orientation. In *Joint Modular Languages Conference* (pp. 59-62). Springer. 10.1007/978-3-540-45213-3_8

Zhu, H., & Zhou, M. (2006). Role-based collaboration and its kernel mechanisms. *IEEE Transactions on Systems, Man and Cybernetics. Part C, Applications and Reviews*, *36*(4), 578–589. doi:10.1109/TSMCC.2006.875726

Chapter 4
Continuum Mechanics for Coordinating Massive Microrobot Swarms:
Self-Assembly Through Artificial Morphogenesis

Bruce MacLennan
University of Tennessee – Knoxville, USA

ABSTRACT

This chapter addresses the problem of coordinating the behavior of very large numbers of microrobots to assemble complex, hierarchically structured physical objects. The approach is patterned after morphogenetic processes during embryological development, in which masses of simple agents (cells) coordinate to produce complex three-dimensional structures. To ensure that the coordination mechanisms scale up to hundreds of thousands or millions of microrobots, the swarm is treated as a continuous mass using partial differential equations. A morphogenetic programming notation permits algorithms to be developed for coordinating dense masses of microrobots. The chapter presents algorithms and simulations for assembling segmented structures (artificial spines and legs) and for routing artificial neural fiber bundles. These algorithms scale over more than four orders of magnitude.

DOI: 10.4018/978-1-5225-5276-5.ch004

GOALS

Although there has been considerable progress in the bulk assembly of nanostructured materials, many future applications of nanotechnology will require the assembly of complex hierarchical systems, structured from the nanoscale up to the macroscale. Examples include future robots, computer systems, and peripheral devices. In some cases, technologies such as 3D printing will permit the fabrication of systems of moderate complexity. However, hierarchical systems that span the full range of scales, from nano to macro, will require self-assembly, at least at the smallest spatial scales.

The fabrication of biological-scale robots illustrates many of the issues: how can we assemble a brain-scale artificial nervous system, high-resolution sensors, effector systems with many degrees of freedom, and so forth? Mammalian brains contain billions of neurons with trillions of interconnections, and it is plausible that artificial neural systems with similar capabilities will require comparable numbers of devices. Mammalian cortex is highly structured and functionally organized; how can we assemble comparable numbers of devices and interconnect them appropriately? For example, the human retina has perhaps 100 million receptors, which compress data into the approximately one million neurons of the optic nerve; we would like to be able to assemble sensors of similar complexity for future robots. Animals behave competently in the physical world by means of detailed proprioceptive, haptic, and other sensory information, which is used to control, in real time, a large number of muscle fibers to achieve fluent, finely controlled, and rapid movement. How can we assemble sensor and effector systems of comparable complexity?

We are investigating the use of swarms of microrobots to assemble such systems, but to do so we need techniques that will scale up to massive (biological) numbers. There is no specific goal number, of course, but we have at least hundreds of thousands in mind, and millions or billions may be required to assemble biological-scale robots. We cannot assume that coordination and communication strategies that work with hundreds, thousands, or even tens of thousands of microrobotic agents will scale up to biological numbers. As is explained in more detail later, we guarantee that our methods will scale by using the same approach that biologists have used for describing the movement of massive numbers of cells: partial differential equations. In effect, we approximate massive numbers of agents by the continuum limit: an infinite number of infinitesimal agents.

We do not know what sorts of microscopic agents will be used for the self-assembly of complex, hierarchical systems; possibilities include microrobots, nanobots, and genetically-engineered microorganisms (effectively organic microrobots). Since there are a variety of possible technologies at various size scales, our goal is an abstract description, independent of the specifics of the agents. That is, we are developing abstract algorithms for self-organization that will produce the desired results so long as the agents have certain basic capabilities (which we are identifying; see *Assumed Capabilities* below).

ARTIFICIAL MORPHOGENESIS

Morphogenesis as a Model

One might legitimately question whether it is even possible to coordinate millions of microscopic agents to assemble complex structures. Fortunately we have an existence proof in *embryological morphogenesis*, which coordinates billions or trillions of cells to assemble a complex, hierarchical body (Nüsslein-Volhard, 2008). Even a relatively simple animal has a large number of distinct tissues, organs, vessels, nerves, etc. that are physically structured in a complex and functional organization. Moreover, multicellular organisms are hierarchically organized from the cellular (and indeed nanoscale) level up to the macroscopic level. Beginning from a single cell, the developing zygote begins to organize itself, establishing poles and layers, and the progressing organization governs future development, so that the microscopic agents (the cells) create the structure that governs their own future behavior. Cells migrate, following chemical gradients, and create forces and pressures that help to shape the tissues. Under the influence of structured signals, cells differentiate into functionally distinct tissues. Thus, the development of the embryo provides an inspiring example of how microscopic agents can coordinate their mutual behavior to self-organize into an immensely complicated structure. Our goal in *artificial morphogenesis* is to mimic these processes for the self-assembly of complex, hierarchical artificial systems by massive microrobot swarms.

Related Work

Artificial morphogenesis has some similarities to *amorphous computing* (Abelson, Allen, Coore, Hanson, Homsy, Knight Jr., Nagpal, Rauch, Sussman, & Weiss, 2000), especially in the earlier stages of morphogenesis, when the agents are in a relatively homogeneous mass. However, as morphogenesis proceeds, the agents arrange themselves into more organized structures, which leads to more structured signaling and interaction, which then leads to further organization. In other words, the goal of artificial morphogenesis is to transform an amorphous or homogeneous initial state into progressively more complex and specific structures. The physical arrangement of the microrobots and their emergent structure of communication and control reinforce each other in an ascending spiral.

Other researchers have recognized the value of morphogenesis as a model (Goldstein, Campbell & Mowry, 2005; Murata & Kurokawa, 2007; Nagpal, Kondacs, & Chang, 2003), and morphogenetic engineering is emerging as a systematic discipline (Bourgine & Lesne, 2011; Doursat, 2008; Doursat, Sayama, & Michel, 2012; Kitano, 1996; Meng & Jin, 2011; Spicher, Michel, & Giavitto, 2005). We believe, however, that these researchers have not applied the principles of morphogenesis systematically enough to scale up to biological numbers. Certainly, in any sort of bio-inspired computing, a crucial issue is how closely to mimic biological processes. This is also the crucial issue in modeling: What is the appropriate level for the model? For artificial morphogenesis, we believe that it is essential to adopt models that obviously apply to very large numbers of robotic agents, comparable to the number of cells in an embryo. This is also essential for the self-assembly of macroscopic structures organized from the nanoscale up.

There has been significant research modeling biological morphogenesis, but sometimes this is too specific to biological systems. For example, the CompuCell3D system models three-dimensional morphogenesis using a cellular Potts model to simulate changes in cell shape during morphogenesis (Cickovski, Huang, Chaturvedi, Glimm, Hentschel, Alber, Glazier, Newman, & Izaguirre, 2005). This is certainly an important issue in biological morphogenesis, but less so in artificial morphogenesis, in which the robotic agents probably do not change shape or do so only in restricted ways. Therefore, we model morphogenesis at a higher, more abstract level, which is more tractable, analytically and computationally. We believe this is the optimal level to express biological-scale morphogenetic algorithms in a way that is relatively independent of specific implementation technology. Embryologists have found it useful for similar reasons (e.g., Forgacs & Newman, 2005; Meinhardt, 1982).

Other related work is described under *Global-to-local Compilation* below.

Morphogenetic Processes

Edelman (1988, p. 17) divides morphogenetic processes into (1) *driving forces* and (2) *regulatory mechanisms*. There are three driving forces: *cell proliferation*, *apoptosis* (programmed cell death), and *cell migration*. Cell proliferation is the mechanism by which embryos grow, but, in the absence of self-reproducing microrobots, it is likely to be less important in artificial morphogenesis. We can often achieve effects similar to cell proliferation either by providing an external supply of microrobots, which migrate to a growth site, or by having a fixed population of microrobots transport passive components to the growth site. One function of cell death in embryological development is to create cavities and passages in a tissue. In principle, we can implement the second driving force, apoptosis, in artificial morphogenesis by programming microrobots to disassemble themselves under appropriate circumstances, but it might be more practical to have them migrate away, or to sculpt cavities by having microrobots remove passive components. In artificial morphogenesis, as in embryogenesis, migration is often guided by chemical signals, for example, following the gradient of a *morphogen*. However, other controllable characteristics of the environment can also guide migration.

The other class of morphogenetic processes, the regulatory mechanisms, comprise *cell adhesion* and *cell differentiation*, both of which have direct analogues in artificial morphogenesis. Microrobots adhere to each other and to passive components in order to change their relative positions and to create permanent or temporary structures. Cells in embryos use adhesion molecules, and some microrobots will use molecular adhesion too, as well as electrostatic, magnetic, and mechanical adhesion (e.g., via latches). The second regulatory mechanism, cell differentiation, is the process by which cells assume different functions by enabling specific regulatory circuits to the exclusion of others. This is one of the central mechanisms of artificial morphogenesis, for different behavioral programs are enabled or disabled by variables within the microrobots. As will become apparent in our examples that follow, this permits identical microrobots to behave differently depending on the morphogenetic context in which they find themselves.

Biologists have identified about a dozen fundamental morphogenetic processes that seem to be sufficient for biological development (Salazar-Ciudad, Jernvall, & Newman, 2003). Therefore they constitute an agenda for artificial morphogenesis (MacLennan, 2010, 2012b). The processes may be classified as (1) *cell autonomous mechanisms*, in which the cells do not interact, (2) *inductive mechanisms*, in which cells change state as a result of communication, and (3) *morphogenetic mechanisms*, which create patterns by rearranging cells without changing cell states.

The cell autonomous mechanisms are involved with cell division, and lead to daughter cells in different states or with different distributions of proteins and other substances. In the absence of self-reproduction, these mechanisms might not be used in robotic implementations of artificial morphogenesis, and they have a limited role even in natural morphogenesis (Salazar-Ciudad et al., 2003). The same effects can be accomplished by other means more suitable to artificial systems.

The inductive mechanisms are based on cell-to-cell communication, which can be implemented by means of diffusible chemicals, by membrane-bound chemicals, or by direct signaling through cell-to-cell junctions. All these are potential inductive mechanisms in artificial morphogenesis as well. Inductive mechanisms are classified as *hierarchical*, in which signaling is primarily unidirectional, or *emergent*, in which cells signal each other reciprocally. The latter leads to more complex self-organization, as for example in reaction-diffusion systems (Meinhardt, 1982; Turing, 1952).

Morphogenetic processes, in the strict sense, rearrange cells without altering their state. One such process is *directed mitosis*, in which cells divide to yield daughter cells with specific orientations. In the absence of cell division, as is most likely the case in artificial morphogenesis, newly placed components will be oriented with respect to those already placed. Another important morphogenetic process is *differential growth*, which creates shape through the mechanical properties, such as the viscosity, elasticity, and cohesiveness, of cells and extracellular matrices (Salazar-Ciudad et al., 2003). *Apoptosis* — programmed cell death — is a third morphogenetic process, which can be used to create form by sculpting cavities. In artificial morphogenesis, microrobots might not "die," but they could migrate away or be disassembled, or the microrobots might eliminate passive material.

Assumed Capabilities

To define the territory between biology and swarm robotics, we have a provisional list of assumed microrobot capabilities:

1. Components may be either active (microrobotic) or passive. Morphogenesis takes place by microrobots self-organizing themselves or organizing passive *material*. The extent to which the microrobots remain part of an assembled structure will depend on their cost and on whether the structure is intended to be able to reconfigure itself (i.e., *artificial metamorphosis*).
2. Microrobots have simple, local sensors (chemical, optical, electrical, mechanical, thermal, etc.). Typically these sensors are capable of responding to the intensity or concentration of a field or substance, or to its gradient.
3. Microrobots have simple effectors for local action (motion, shape, adhesion) and for signal generation (chemical, electrical, mechanical, etc.). Microrobots move primarily by simple mechanical interactions with their immediate surroundings (adhesion, traction, pressure, torque, etc.).
4. Long-range communication is accomplished in a variety of ways. One is direct robot-to-robot signaling, which allows a signal to propagate in a directed or undirected way through a sufficiently dense swarm of microrobots. Long-range signaling can also be accomplished by diffusion, which can be active or passive. Passive diffusion is a result of Brownian motion of microrobots or passive components (such as signaling molecules), and active diffusion is a result of microrobots moving under their own power, but randomly. Finally, long-range signaling can be implemented by means of more directed processes: microrobots can move from one place to another, and thereby represent information by their presence, or they can transport passive information-bearing objects.
5. The behavior of microrobots is governed by simple regulatory circuits, which need not be electrical. For example, biological microrobots may use genetic regulatory circuits. In general, regulation will be more like analog control than

digital computation (Teo, Woo, & Sarpeshkar, 2015). Therefore, most often our programs for microrobot control will take the form of differential equations.

6. Microrobots might be self-reproducing or not, but, to ensure wider applicability, we focus on processes that can be accomplished without self-reproduction. For example, where biological morphogenesis uses cell proliferation, artificial morphogenesis might use microrobots to transport passive components to a growth area. (See the next section for more on this issue.)
7. Microrobots make use of ambient energy and/or distributed fuel. In general, they will not be able to store enough energy for prolonged operation, so they will make use of thermal, optical, chemical, electrical, mechanical, or other ambient and distributed energy sources. Metabolism may be used by organic microrobots (genetically engineered microorganisms). Hybrid agents might combine microrobots with living cells for energy or other tasks more easily accomplished by living cells.

Non-Biological Implementations

Obviously some things must be done differently in non-biological systems from the way they are accomplished in embryos. For example, tissue growth and some aspects of biological morphogenesis are a result of cell proliferation, but in the likely absence of self-reproduction (cell division), artificial morphogenesis must accomplish them differently. One way is to have an external supply of microrobots migrate to the growth site, where they become a permanent part of the self-organizing system. Another is to have microrobots transport passive components to the growth site and to insert them in the tissue. Even with these alternative mechanisms, some biological morphogenetic processes will have to be adapted for artificial systems. For example, in embryos cell proliferation may take place in the interior of a cell mass, which is difficult to mimic when new components are provided externally. In these cases, we may have to arrange alternate ways of getting new components to the growth sites (e.g., through open passages).

Communication and Coordination Mechanisms

As previously remarked, the primary communication and coordination mechanisms are by means of contact, diffusion, and movement. Microrobots may affect and sense components (active or passive) in their immediate vicinity and thereby transfer information. For longer-distance "broadcasting" they may produce substances or disturbances in the medium, which disperse by diffusion or wave propagation. Most importantly, microrobots can move within the medium, thereby conveying both information and control by means of their presence or by means of the objects they transport. In general, we do not assume any long-range communication or coordination mechanisms except what might be provided by macroscopic external fields (e.g., gravitational, electric, magnetic). Our approach to the coordination of microrobot swarms does not require establishment of any sort of global coordinate system or external patterning.

MORPHOGENETIC PROGRAMMING

Soft Matter

Embryological morphogenesis takes place in the regime of *soft matter*, or *viscoelastic material* (de Gennes, 1992). These are materials which stretch, bend, and fold when subjected to weaker forces, and flow viscously when subjected to stronger ones, which is the way tissues behave in the developing embryo. These viscoelastic properties are fundamental to the creation of embryonic form and structure. Similarly, application of these processes in artificial morphogenesis, involving very large numbers of microrobots and passive materials connected more or less strongly, suggests that these self-assembling structures be treated as soft matter ("tissues").

Continuum Mechanics Framework

The appropriate mathematical framework for describing viscoelastic materials is continuum mechanics expressed in partial differential equations (Beysens, Forgacs, & Glazier, 2000; Forgacs & Newman, 2005; Meinhardt, 1982; Taber, 2004) Therefore, an artificial morphogenetic system is described as one or more three-dimensional *bodies* (which might be fluids) continuously evolving in three-dimensional space as a result of external and internal forces. Although the bodies may be composed of discrete elements (e.g., molecules, cells, or microrobots), we adopt a level of abstraction that permits bodies to be treated as continua. The dynamics of the system can be described from the perspective of an external observer (an *Eulerian* or *spatial* frame of reference) or from the perspective of an infinitesimal volume element of the material (a *Lagrangian* or *material* frame of reference). The latter is a more agent-oriented perspective, suitable for programming microrobots, which biologists have found valuable as well (Bonabeau, 1997). These infinitesimal volumes of material are called *parcels* or *material particles*. (Mathematical details can be found in prior reports: MacLennan, 2010, 2011, in press.)

To describe change mathematically, we can focus either on a fixed location $\mathbf{p}$ in space and describe how some quantity $q(\mathbf{p}, t)$ changes at that location, or we can focus on a fixed particle *P* as it moves through space and how that property *q*(*P*, *t*) of the particle changes. The former is called a *spatial derivative*, written $\partial q \,/\, \partial t$ and the latter a *material* or *substantial derivative*, written $\mathrm{D}q \,/\, \mathrm{D}t$. The location **p** of a particle is a continuous function of time, and so the chain rule allows us to express the substantial derivative in terms of the spatial derivative:

$$\frac{\mathrm{D}q\left(P,t\right)}{\mathrm{D}t} = \frac{\partial q\left(\mathbf{p},t\right)}{\partial t} + \mathbf{v}\cdot\nabla q,$$

where **v** is the velocity field of the particles. That is, the substantial rate of change is the sum of the spatial rate of change and the convective rate of change. In particular, acceleration may be expressed either as a property of particles (a more agent oriented, Lagrangian perspective) or a property at a particular location through which there is a flux of particles (a spatial or Eulerian perspective). If v_i .are the Cartesian components of a particle's velocity **v**, then the Cartesian components of its acceleration are the substantial derivatives of these components:

$$A_i = \mathrm{D}v_i \,/\, \mathrm{D}t = \partial v_i \,/\, \partial t + \mathbf{v}\cdot\nabla v_i.$$

If we let $\nabla\mathbf{v}$ be a second-order tensor [with Cartesian component $\left(\nabla\mathbf{v}\right)_{ij} = \partial v_i \,/\, \partial x_j$, then the substantial acceleration can be expressed

$$A = \partial v \,/\, \partial t + v\cdot\nabla v = a + v\cdot\nabla v,$$

where **a** is the spatial acceleration field. Thus the substantial acceleration is the spatial acceleration plus the convective acceleration.

Continuum Programming Notation

We have developed an experimental programming notation for expressing morphogenetic algorithms at a level suitable for control of massive swarms of microscopic robots and for modeling their behavior (MacLennan, 2010, 2011, 2012b). Primarily, it is a notation for programming in stochastic partial differential equations, with suitable means for definition and initialization of variables. Our goal is that it be directly executable as a simulation language, but that it also serve as a specification language for designing microrobots and passive materials that can be applied to artificial morphogenesis.

Change Equations

Behavior is described by *change equations*, which can express continuous, discrete, and qualitative change. The notation $ĐX = F\left(X, Y, \ldots\right)$ can be interpreted either as a PDE (partial differential equation), $\partial X / \partial t = F\left(X, Y, \ldots\right)$ or as a temporal finite difference equation, $\Delta X / \Delta t = F\left(X, Y, \ldots\right)$. This systematic ambiguity, which permits realization as either a discrete- or continuous-time system, is respected by the formal rules of manipulation for the notation. More precisely, change equations are treated as *dynamic equations on time scales* (Bohner & Peterson, 2001; Agarwal, Bohner, O'Regan & Peterson, 2002), in which in our case the time scale is either $\mathbb{R}$ the real numbers, or $\left(\Delta t\right)\mathbb{Z} = \left\{\ldots, -2\Delta t, -\Delta t, 0, \Delta t, 2\Delta t, \ldots\right\}$. Space can be similarly treated as ambiguously discrete or continuous by defining partial differential equations over time scales (Ahlbrandt & Morian, 2002), but treating it continuously helps to ensure that our coordination strategies scale to truly massive robot swarms.

Sometimes it is convenient to break a long change equation into parts, and so we allow it to be expressed in several *extended equations*, which might be textually separated in a program. For example, an equation of the form $ĐX = F\left(X, Y, \ldots\right) + G\left(X, Y, \ldots\right) - H\left(X, Y, \ldots\right)$ could be broken into partial equations:

$$
\begin{aligned}
&ĐX+ = F(X, Y, ...),\\
&ĐX+ = G(X, Y, ...),\\
&ĐY- = H(X, Y, ...)
\end{aligned}
$$

This notation was first applied to morphogenesis by Kurt Fleischer (1995, p. 20).

Due to many stochastic factors, morphogenetic processes, both natural and artificial, must be robust. Therefore, in many cases, a precise functional dependence is not so important as whether one quantity tends to increase or decrease another. Embryologists express these relationships in *influence diagrams*, which show how one quantity promotes or represses another. Our morphogenetic programming notation has a similar concept, a change *regulation* (as opposed to a change *equation*). For example, $ĐX \simeq X, -Y, Z$ means that the change of *X* is promoted (positively regulated) by *X* and *Z*, but repressed (negatively regulated) by *Y*. More precisely, this regulation is interpreted as an equation $ĐX = F\left(X, -Y, Z\right)$ in which *F* is an unspecified function that is monotonically increasing in each of its arguments. This allows us to specify algorithms that omit details that are irrelevant or that will be determined in a specific implementation.

For convenience we use a conditional notation to allow a threshold to gate the influence of a quantity; it is defined:

$$\left[x > \vartheta\right] = \begin{cases} 1, & \text{if } x > \vartheta, \\ 0, & \text{otherwise.} \end{cases}$$

More complex conditions have the obvious meaning; for example $\left[x > \vartheta \wedge y > \varphi\right] = \left[x > \vartheta\right] \times \left[y > \varphi\right]$. Conditions are one of the principal tools for enabling or disabling particular behavioral equations, which fulfills purposes similar to cell differentiation in biological morphogenesis. A typical application is to enable or block some process if a quantity is above a threshold.

Substances

Analogous to classes in object-oriented languages, we have *substances* with properties. For example, the following **substance** declaration defines “morphogen” to be a substance with several fixed properties (diffusion rate, decay rate, etc.) and a concentration that can vary over a region of space:

substance morphogen:
scalars:
D_a || diffusion rate
κ_a || production rate
τ_a || decay time constant
ϑ_a || threshold
scalar field a || concentration
behavior:
$Ða = \left[A > \vartheta_a\right] \kappa_a S\left(1 - a\right) + D_a \nabla^2 a - a / \tau_a$

The **behavior** part defines the dynamics of the substance’s properties by means of formulas, primarily change equations and regulations.

Like classes in object-oriented languages, substances can be subclasses of other substances, and can inherit or override variables and behaviors defined in their superclasses. For example, a substance composed of a swarm of microrobots that follow the morphogen gradient could be defined:

substance swarm **is** morphogen **with**:
vector field v || velocity field

scalar field M $\|$ microrobot concentration
scalar μ $\|$ microrobot mobility
behavior:
$\mathbf{v} = \mu \nabla a$
$ĐM = -\nabla \cdot M\mathbf{v}$

A substance can be a subclass of several different substances, thereby permitting it to inherit the properties of those substances. (Additional information can be found in prior publications: MacLennan, 2010, 2011, 2012b).

It is useful to distinguish two kinds of substances: *physical substances*, whose properties are relatively fixed, and *controllable substances*, whose properties are relatively alterable; they correspond roughly to hardware and software. For example, both passive components and active components (agents) will have physical properties that are relatively fixed, such as mass, size, and (for agents) a complement of sensors and actuators. These properties are described by a physical substance definition, which describes, for example, the flux of particles when subjected to a force. Agents, however, have in addition properties that are relatively alterable. For example, the behavior of microrobots can be controlled by an analog or digital control program, and genetically-engineered microorganisms can be programmed by modifying their genetic regulatory mechanisms, which have many similarities to analog control (Teo, Woo & Sarpeshkar, 2015). We coordinate swarms of microrobots primarily by defining their controllable behavior operating within the constraints of the physical properties of the microrobots and the passive components. The physical properties are specified by a physical substance definition, and the programmable behavior is specified by a controllable substance, which it typically a subclass of the physical substance. A behavior may be defined by an extended equation, with one partial equation in a physical substance describing its fixed behavior (e.g., how it responds to a motive force), and another partial equation in a controllable substance describing its programmed behavior (e.g., the motive force it exerts in particular situations). As in object-oriented programming, an unspecified *virtual* property in the physical substance may be defined in the controllable subclass substance. At this time, however, we do not distinguish physical and controllable substances in the morphogenetic programming notation, since it is more relevant to the physical interpretation of the notation than to its formal applications.

Bodies

Specific instances of substances are called *fields*, *tissues*, or *bodies* and are created by a **body** declaration, which gives initial values to all its variables. For example, the following defines a spherical region of morphogen with all of the morphogen concentrated initially near the center:

body MorphogenField **of** morphogen:
for $\|\mathbf{p}\| \leq 1$
$D_a = 0.1$
$\kappa_a = 0.2$
$\tau_a = 0.01$
$\vartheta_a = 0.6$
for $\|\mathbf{p}\| \leq 0.001$ $a = 1$ || concentration in center
for $\|\mathbf{p}\| > 0.001$ $a = 0$ || concentration in remainder

The automatically declared vector variable **p** refers to an arbitrary location in the body. The **for**-declarations then define the spatially distributed initial values of a body of the specified substance. For example, D_a has a value of 0.1 throughout a sphere of radius 1 centered at the origin. The morphogen concentration is $a = 1$ within a smaller sphere (radius 0.001), but $a = 0$ outside of it. The intention of **body** declarations is that the initial distributions be simple so that they can be physically prepared.

Global-to-Local Compilation

An important issue in swarm robotics—indeed, in any application of emergent phenomena—is *global-to-local compilation*, that is, the translation of some desired global behavior into the required behavior of its constituent elements or agents (Yamins, 2005, 2007; Yamins & Nagpal, 2008). One approach applies classical control theory to swarm robotics (Feddema, Lewis & Schoenwald, 2002; Gazi & Passino, 2003), but its use is limited by assuming global or restricted communication and by neglecting realistic conditions for large, microscopic swarms, such as stochastic effects and non-determinism. Formal methods likewise have been limited to relatively simple collective behaviors and shapes (Kornienko, Kornienko & Levi, 2005b; Winfield, Sav, Fernández-Gago, Dixon, & Fisher, 2005; Soysal & Şahin, 2007). Martinoli, Lerman, and their collaborators have pursued an alternative approach based on chemical rate equations (Martinoli, 1999; Lerman & Galstyan, 2002; Martinoli, Easton & Agassounon, 2004; Lerman, 2004; Lerman, Martinoli & Galstyan, 2005; Correll & Martinoli, 2006; Correll, 2007), but these do not address behavior at the agent level and are unable to deal with detailed spatial patterning as required for artificial morphogenesis. Another class of approaches uses Brownian motion as a way of relating macroscopic and microscopic behavior. This includes work by Helbing, Schweitzer, Keltsch, and Molnar (1997), Vicsek, Czirok, Ben-Jacob, Cohen, and Shochet (1995), Schweitzer (2002, 2003; Schweitzer, Lao & Family, 1997), Hogg (2006), and Hamann (2010). These approaches are limited to the formation of relatively simple aggregation patterns determined, for example, by potential functions and gradients. In summary, previous approaches to global-to-local compilation have not demonstrated their adequacy to artificial morphogenesis.

In artificial morphogenesis, global-to-local compilation fulfills a specific role in the development of a morphogenetic program. We generally begin with a system of partial differential equations that describe the movement of a massive swarm to assemble the desired structure. Often, these PDEs are derived from biological descriptions of natural morphogenetic processes. Generally the mathematics will use an Eulerian reference frame, since we take a perspective from outside the embryo or morphogenetic system. Second, we translate the equations into a Lagrangian (material) frame, which takes the perspective of individual particles moving *en masse*, that is, the perspective of agents moving through space, interacting with other agents and responding to their environment. Often, these equations can be separated into two components: a physical substance, which captures the physical properties of the agents, and a controllable substance, which describes its programmable behavior (sensing, control, actuation). In the continuum mechanics framework, the Lagrangian PDEs describe the motion of infinitesimal particles or parcels of controllable substance. The control equations for these infinitesimal agent particles must be translated to control equations for microrobots (or microorganisms) of finite size, mass, etc. (Whereas most approaches to swarm robotics have the problem of scaling *up* to a large number of agents, we have the complementary, but easier problem of scaling *down* from an infinite number of agents.)

The translation of the control equations from infinitesimal particles to finite agents is an ongoing subject of research, but *smoothed particle hydrodynamics* (SPH) is a promising approach. It is a mesh-free particle simulation method for continuous media (Gingold & Monaghan, 1977; Liu & Liu, 2003), and by associating appropriate kernel functions with the particles, it provides a useful bridge between continua and finite ensembles of particles. It has been applied already to the control of robot swarms of modest size (Bandala & Dadios, 2016; Perkinson & Shafai, 2005; Pac, Erkmen & Erkmen, 2007; Pimenta, Mendes, Mesquita, et al., 2007; Pimenta, Pereira, Michael, et al., 2013; Song, Lipinski & Mohseni, 2017), and we anticipate that it will be useful in artificial morphogenesis as well.

EXAMPLE: SEGMENTATION

To further develop the technique and notation for morphogenetic programming, we have investigated several examples of morphogenesis. In each case we mimic processes that occur in biological morphogenesis, but we apply them in different contexts, motivated primarily by robotics. Two such applications are described here.

Segmentation

In vertebrate animals, the number of vertebrae is characteristic of the species. How does an embryo generate such a precise number of segments (e.g., 33 for humans, 55 for chickens)? In vertebrate embryos, segmentation takes place through the clock-and-wavefront process (Cooke & Zeeman, 1976; Dequéant & Pourquié, 2008), and this process can be applied to similar ends in artificial morphogenesis. Therefore, our first example explores the assembly of segmented body parts, such as an artificial spine and legs. (We present new results here; more detailed background information can be found in prior publications: MacLennan, 2012a, 2012b, in press.)

We begin with the spinal axis. The assembly is initialized with a small segment of *rostral tissue* ($S = 1$) at the head and a small segment of *caudal tissue* ($T = 1$) at the end (the *tail bud*). These are two differentiated states of the "medium" substance, and so $M = 1$ throughout the tissue. The growth duration t_G is regulated by a quantity G, which is initialized to G_0 and decays according to $ĐG = -G / \tau_G$. Growth continues so long as G is greater than a threshold ϑ_G and therefore $t_G = \tau_G \ln\left(G_0 / \vartheta_G\right)$. For convenience, we usually choose our units so that $G_0 = 1$ and $\vartheta_G = 1/e$ with the result that $t_G = \tau_G$.

In an embryo this growth is a result of cell division; in artificial morphogenesis it is a result of external addition of microrobots, but the specific process is left open in this model. Therefore, we specify only that the microrobots in the tail bud are initialized in a direction **u** pointing away from the head and that they move in that direction at a rate r so long as growth continues. Therefore the effective velocity of these microrobots is $\mathbf{v} = \left[G > \vartheta_G\right] r\mathbf{u}$ and the resulting extension of the body is $r\tau_G \ln\left(G_0 / \vartheta_G\right)$ typically $r\tau_G$. The movement of the mass of microrobots constituting the tail bud is given by the negative divergence of the microrobot flux $T\mathbf{v}$:

$$ĐT = -\nabla \cdot T\mathbf{v} = -T\nabla \cdot \mathbf{v} - \mathbf{v} \cdot \nabla T.$$

The spine grows linearly in the caudal direction through the addition of undifferentiated tissue ($M=1, S=0$) between the head and tail buds, $\text{Đ}M = rT / \lambda_{\text{TB}}$ where λ_{TB} is the length of the tail bud.

Two morphogens and the wavefront signal can be used to control the number and length of the segments. A *caudal morphogen* diffuses from the tail bud, where it is produced up to saturation:

$$\text{Ð}C = D_C \nabla^2 C - C / \tau_C + \kappa_C T \left(1 - C\right)$$

The *rostral morphogen* diffuses similarly from already differentiated tissue ($S = 1$) at the head end:

$$\text{Ð}R = D_R \nabla^2 R - R / \tau_R + \kappa_R S \left(1 - R\right)$$

Under steady-state conditions, the concentration of R on the axis at a distance x from fully differentiated tissue is:

$$R\left(x\right) = \frac{\kappa_R}{4\pi D_R x} \exp\left(-\frac{x}{\sqrt{D_R \tau_R}}\right)$$

Since the tail bud is moving with a velocity r, C obeys an advection-diffusion equation, and the concentration of C on the axis at a distance x behind the tail bud is:

$$C\left(x\right) = \frac{\kappa_C}{4\pi D_C x} \exp\left[-x \frac{\sqrt{r^2 + 4D_C / \tau_C} + r}{2D_C}\right]$$

These equations allow the computation of R and C thresholds (R_{upb} C_{upb} respectively) to define the region in which a new segment will differentiate.

While growth continues ($G > \vartheta_G$, microrobots in the tail bud generate a pacemaker or clock signal:

$$\text{Ð}K = \omega^2 L$$
$$\text{Ð}L = [G > \vartheta_G] K.$$

When the clock is in the correct phase ($K > \vartheta_K$, the tail tissue generates a pulse of the *segmentation morphogen* α which will propagate in a wave toward the head. This pulse is represented by $\psi = \left[G > \vartheta_G \wedge K > \vartheta_K\right] T$. The microrobot swarm constituting the undifferentiated tissue functions as an excitable medium, and segmentation morphogen concentration above a threshold ($\alpha > \vartheta_\alpha$ causes it to generate a new pulse of the morphogen, provided that the tissue is not in its refractory period (represented by $\rho < \vartheta_\rho$. Therefore, this pulse of α is represented by $\phi = \left[\alpha > \vartheta_\alpha \wedge \rho < \vartheta_\rho\right] M$. The generation and propagation of the segmentation morphogen and the decay of the refractory factor are described by the change equations:

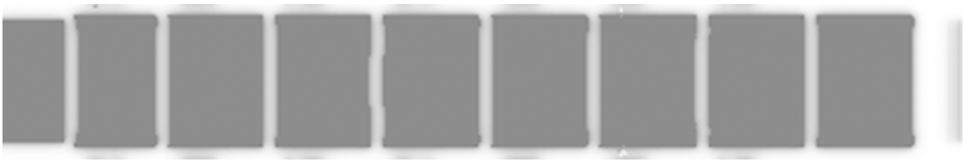
$$\text{Ð}\alpha = \phi + \psi + D_\alpha \nabla^2 \alpha - \alpha / \tau_\alpha,$$
$$\text{Ð}\rho = \phi - \rho / \tau_\rho.$$

The new segment is formed between already segmented rostral tissue ($S \approx 1$ and the tail bud, a region identified by both rostral and caudal morphogens below threshold (R_{upb} C_{upb} respectively), as the wave of segmentation morphogen passes through in a rostral direction (Figure 1):

$$\text{Ð}S = \left[\alpha > \alpha_{\text{lwb}} \wedge R < R_{\text{upb}} \wedge C < C_{\text{upb}}\right] c_S + \kappa_S S\left(1 - S\right).$$

Segment differentiation is triggered by the first term, which is a transient pulse, and the second term continues its logistic growth to saturation (completely differentiated, $S = 1$).

Figure 1. Two-dimensional simulation of spinal segmentation in progress. In addition to the initial head segment on the far left, eight complete segments have assembled. The haze around these segments represents the concentration of rostral morphogen diffusing from them. The band on the far right represents concentration of caudal morphogen diffusing from the tail tissue (not shown). The image is restricted to the spinal region.



The length λ of the segments is controlled by the ratio of the tail growth rate *r* and the clock frequency ω in cycles per unit time: $\lambda = 2\pi r \,/\, \omega$ since this is the distance the tail bud moves in one clock cycle. The number of segments *n* is determined by the product of the frequency and duration of growth t_G $n = t_G \omega \,/\, 2\pi$. Therefore, the number and length of the segments is controlled jointly by *r*, t_G and ω.

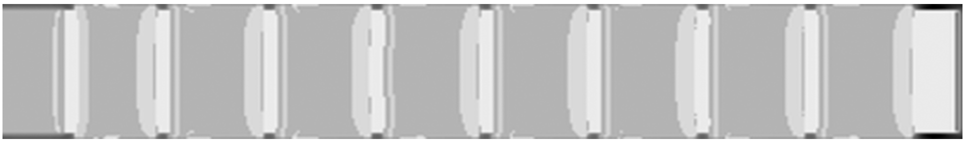
We would like to be able to control the placement of legs within each segment, but this requires that we be able to distinguish the anterior and posterior ends of the segments. In particular, we want to polarize the segments by having the anterior ends of differentiated segment tissue (*S* = 1) further differentiate into anterior tissue (*A* = 1), and the posterior ends into posterior tissue (*P* = 1); see Figure 2. This is accomplished by making the differentiated *S* tissue sensitive to appropriate values of the rostral and caudal morphogens when a subsequent segmentation wave passes through:

To determine the placement of the legs, we use anterior (*a*) and posterior (*p*) morphogens, which accumulate up to saturation in tissue where the anterior and posterior microrobots are sufficiently dense. The *a* and *p* morphogens then diffuse from *A* and *P* regions, respectively. This behavior is described as follows:

$$\text{Đ}a = [A > \vartheta_A]\kappa_a S(1 - a) + D_a \nabla^2 a - a \,/\, \tau_a,$$
$$\text{Đ}p = [P > \vartheta_P]\kappa_P S(1 - p) + D_a \nabla^2 p - p \,/\, \tau_p.$$

Legs grow from the surface of the spinal segments, but it is easy for microrobots to determine whether they are on the outer surface of the spinal tissue (and set *E* = 1) as opposed to the interior (*E* = 0). For example, a microrobot can estimate the local population density in *S* by *quorum sensing*. The local density is near its

Figure 2. Two-dimensional simulation of differentiation of anterior and posterior tissue. Within each segment, the narrow band at the anterior (left) end is anterior (A) tissue, and the wider band at the posterior (right) end is posterior (P) tissue. Between the posterior tissue of one segment and the anterior tissue of the next is undifferentiated inter-segmental tissue (M=1, S=0). The figure is restricted to the spinal region.



maximum ($S = 1$ in our units) in the interior, decreases nearer to the outer surface, and is zero outside of the spine. The quorum sensing process can be expressed by a convolution, $K \otimes S$ where

$$(K \otimes S)(\mathbf{p}) = \int_{\mathbb{R}^3} K(\mathbf{p}-\mathbf{q})S(\mathbf{q})\mathrm{d}\mathbf{q},.$$

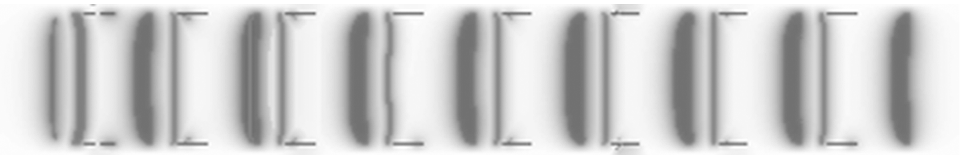
and K is a kernel representing a microrobot's perceptual range. For example, if the microrobots have a certain sensor range, then K will be a sphere of that radius. On the other hand, quorum sensing can be accomplished by having the microrobots emit a slowly diffusing, rapidly degrading signal, which can be sensed to estimate the local population density. In this case K is a radial function that decreases exponentially. Regardless of how quorum sensing is implemented, being on the surface of the spine is indicated by $E = 1$, where $E = [S_{\text{lwb}} < K \otimes S < S_{\text{upb}}]$.

The position of the "imaginal disks" (indicated by $I = 1$), where the legs will be assembled, is controlled by the a and p morphogens and the edge signal E (see Figure 3). The morphogens must be in appropriate ranges to control the anterior-posterior position of legs within each spinal segment, and of course the disks form only on the surface:

$$ÐI = \left[a_{\text{upb}} > a > a_{\text{lwb}} \wedge p_{\text{upb}} > p > p_{\text{lwb}}\right]\kappa_I ES\left(1 - I\right).$$

The clock-and-wavefront process can also be applied to the generation of segmented legs; indeed the same morphogens can be used by changing the clock frequency

Figure 3. Two-dimensional simulation of placement of imaginal disks. The wide bands represent the concentration of anterior (a) and posterior (p) morphogens in each spinal segment. The narrow lines represent the imaginal disk tissue (seen edge on) on the surface of the spine; their position is determined by the concentrations of the anterior and posterior morphogens. A line of segmentation morphogen (α can be seen propagating through the posterior region of the third segment. The figure is restricted to the spinal region.



and initial *G* value. To assemble appendages of length λ_{A} we use an initial *G* value $G_{\mathrm{A}} = \vartheta_G \exp\left(\lambda_{\mathrm{A}} / r\tau_G\right)$. To assemble *n* leg segments, we use a frequency $\omega_{\mathrm{A}} = 2\pi nr / \lambda_{\mathrm{A}}$. However, this process must be correctly initialized by generating properly oriented terminal tissue analogous to the tail bud in spinal generation. This initialization can be triggered by rapid differentiation of the imaginal disk, $\text{Đ}I > \vartheta_{\mathrm{DI}}$. *I* increases logistically with a maximum rate $\kappa_I / 2$. Therefore, the length of the initialization period is governed by $\kappa_I / 2 - \vartheta_{\mathrm{DI}}$. During the initialization phase the microrobots in the imaginal disks differentiate into terminal tissue, $\text{Đ}T = [\text{Đ}I > \vartheta_{\mathrm{DI}}]\kappa_{\mathrm{AT}}I\left(1 - T\right)$ and reorient themselves in an outward direction (down the gradient of *S*):

$$\text{Đ}\mathbf{u} = \left[\text{Đ}I > \vartheta_{\mathrm{DI}}\right]\kappa_{\mathrm{A}}I\left(-\frac{\nabla S}{\left\|\nabla S\right\|} - \mathbf{u}\right).$$

The same signal triggers rapid resetting of the growth and clock parameters:

$$\text{Đ}G{+} = [\text{Đ}I > \vartheta_{D1}]\kappa_{G0}I(G_A - G),$$
$$\text{Đ}\omega{+} = [\text{Đ}I > \vartheta_{D1}]\kappa_{G0}I(\omega_A - \omega).$$

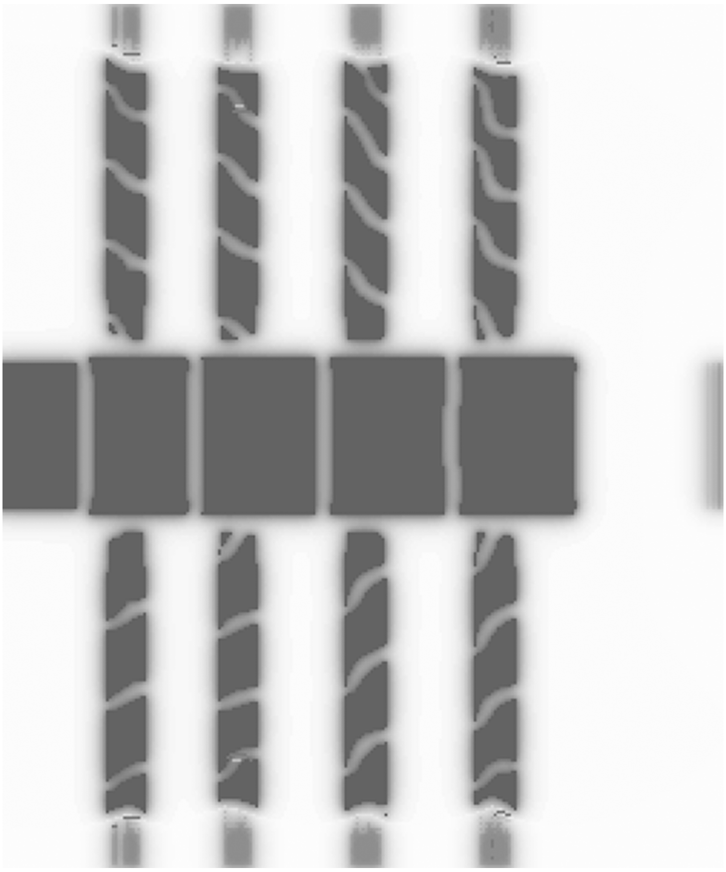
After this initialization, the equations defining the clock-and-wavefront process take over, but generating leg segments as determined by the G_{A} and ω_{A} parameters. See Figure 4.

There is a complication that must be avoided. If the segmentation of the legs follows exactly the same process as the segmentation of the spine, then the leg segments will develop their own imaginal disks and grow their own little “leglets.” To prevent this, differentiation of the spinal imaginal disks suppresses generation of the *a* and *p* morphogens in the imaginal disks, which prevents generation of imaginal disks on the legs:

$$\text{Đ}\kappa_a = -\kappa_a I / \tau_I,$$

$$\text{Đ}\kappa_p = -\kappa_p I / \tau_I.$$

Figure 4. Two-dimensional simulation of growth and segmentation of first four pairs of legs. In this simulation, spinal growth has been limited to four segments. The legs also have four segments completed or under assembly (although in some cases a fifth, rudimentary segment has formed). The joints are angled due to the continuing diffusion of rostral morphogen from the spinal segments.



EXAMPLE: NEURAL ROUTING

As distant functional areas of the brain must be connected, so must the parts of an artificial nervous system. In embryos, a *growth cone* at the end of a growing axon follows chemical signals to its destination. We can use a similar approach, either by using multiple diffusible attractants to ensure that different, simultaneously developing fibers find their destinations correctly, or by creating one fiber at a time, so there is no chance of interference. For the sake of this example, we use the latter, simpler approach. To illustrate the procedure for developing an algorithm for coordinating a massive robot swarm, we will solve this problem in three phases.

In the first phase a growth cone is represented by a single microrobot that generates a fiber from an origin to a destination (see MacLennan, 2012b for additional detail). To establish the pathway we have the goal region ($G = 1$) produce a diffusible attractant A, which also decays to avoid saturation of the space through which the fibers pass:

$$\text{Ð}A = D_A \nabla^2 A - A / \tau_A + \kappa_A G \left(1 - A\right).$$

The growth-cone microrobot departs from its origin following the attractant gradient and creates the fiber (represented as a concentration of $P \approx 1$ in its wake. In the process of finding its way to its destination, we do not want the new fiber to collide with already established fibers. One way to accomplish this is to have existing fibers emit a repellant R to keep new fibers at a safe distance:

$$\text{Ð}R = D_R \nabla^2 R - R / \tau_R + \kappa_R P \left(1 - R\right).$$

The growth cone then follows the gradient of the difference between the attractant and repellant. However, we want the growth cone to move at a constant rate in spite of the fact that the concentration of the gradient decreases exponentially with distance from the goal. Therefore we set the velocity **v** of the growth cone proportional to the versor of the gradient (i.e., to the normed gradient): $\mathbf{v} = r \nabla M / \left\| \nabla M \right\|$ where $M = A - R$. (If the gradient is zero then the velocity is unchanged.) This velocity then determines the change in the growth cone's position, $\text{Ð}\mathbf{p} = \mathbf{v}$. (Every substance has an intrinsic vector field **p**, the spatial location of every particle.) Therefore the growth cone substance is defined as a subclass inheriting the attractant and repellant substances:

substance growth_cone **is** attractant **and** repellant **with**:
scalar r || cone migration rate
vector field v || velocity field
scalar field M || net attraction
scalar field P || fiber material
order-2 **field** σ || diffusion tensor
order-1 **random W** || random vector
scalar κ_P || fiber deposition rate
behavior:
$M = A - R$

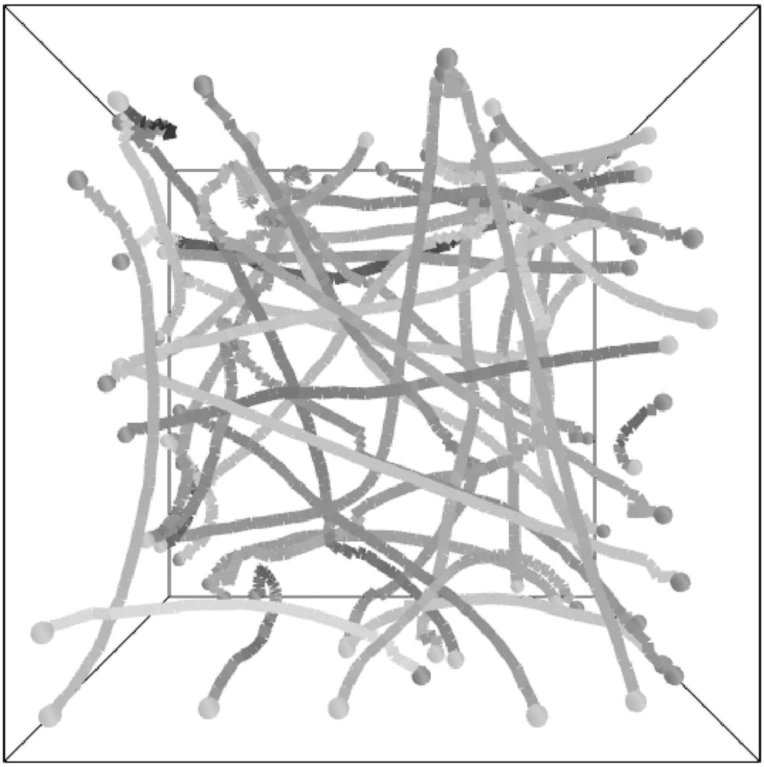
$$\mathbf{v} = r\nabla M \,/\, \left\|\nabla M\right\|$$
$$\text{Đ}\mathbf{p} = \mathbf{v} + \sigma\text{Đ}\mathbf{W}$$
$$\text{Đ}P = \kappa_P\left(1 - P\right)$$

Because $\text{Đ}\mathbf{W}$ is a normally-distributed random vector (as explained in MacLennan, 2011, 2012a, 2012b), the term $\sigma\text{Đ}\mathbf{W}$ generates a small random perturbation that keeps the growth cone from getting stuck. The fiber material *P* is deposited by the growth-cone microrobot: $\text{Đ}P = \kappa_P\left(1 - P\right)$

Figure 5 shows a typical result of simulating the sequential generation of 40 neural fibers between randomly generated origins and destinations. The fibers occupy about 4.5% of the interior space without collisions.

For the second phase of algorithm development, we extend this solution from a single agent generating a fiber to a *cohort* of microrobots, which creates a fiber bundle by departing from the source region and following an attractant gradient to

Figure 5. Simulation of path routing. Forty neural fibers are routed between randomly chosen origins and destinations on four surfaces.



the destination. Each microrobot creates a nerve fiber in its wake. This creates a potential problem, since the streamlines defined by the gradient converge on the destination, so the diameter of the fiber bundle could decrease as it approaches its destination. We can keep its diameter constant by requiring the microrobots to keep within a certain distance range from each other. This suggests the use of a modified flocking algorithm to coordinate the movement of the microrobots (Reynolds, 1987; Spector, Klein, Perry & Feinstein, 2005).

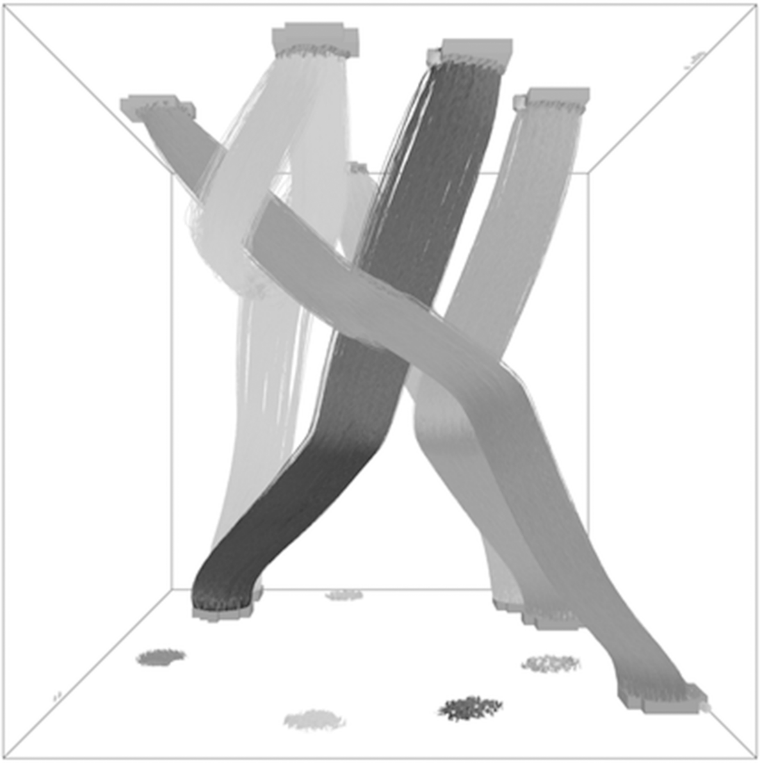
As before, an attractant diffuses from the destination and decays, but instead of requiring a repellant, we assume that the microrobots can detect the proximity of an existing fiber bundle in some unspecified way. The acceleration of a microrobot is determined by the weighted sum of six "urges":

$$\mathbf{a} = w_d\mathbf{u}_d + w_a\mathbf{u}_a + w_c\mathbf{u}_c + w_s\mathbf{u}_s + w_v\mathbf{u}_v + w_w\mathbf{u}_w.$$

The *destination urge* $\mathbf{u}_d$ is proportional to the gradient of the attractant. The *avoidance urge* $\mathbf{u}_a$ steers away from already generated fibers within a specified sensor range, thus avoiding collisions between fibers. The *center urge* $\mathbf{u}_c$ is directed toward the centroid of the positions of the robot's "flock mates" (robots within a certain specified radius, if any); this tends to keep the cohort cohesive. The *spacing urge* $\mathbf{u}_s$ steers away from flock mates within a specified critical distance, and thus prevents the cohort from becoming too compact. It is the balance between the center and spacing urges that regulates the size of the cohort, and hence the diameter of the fiber bundle. The *velocity urge* $\mathbf{u}_v$ is the average velocity of the flock mates, which encourages the robots to move in the same direction. The *wander urge* $\mathbf{u}_w$ is random, which helps prevent the cohort members from getting stuck. The resulting acceleration and velocity are both limited to be below specified maxima.

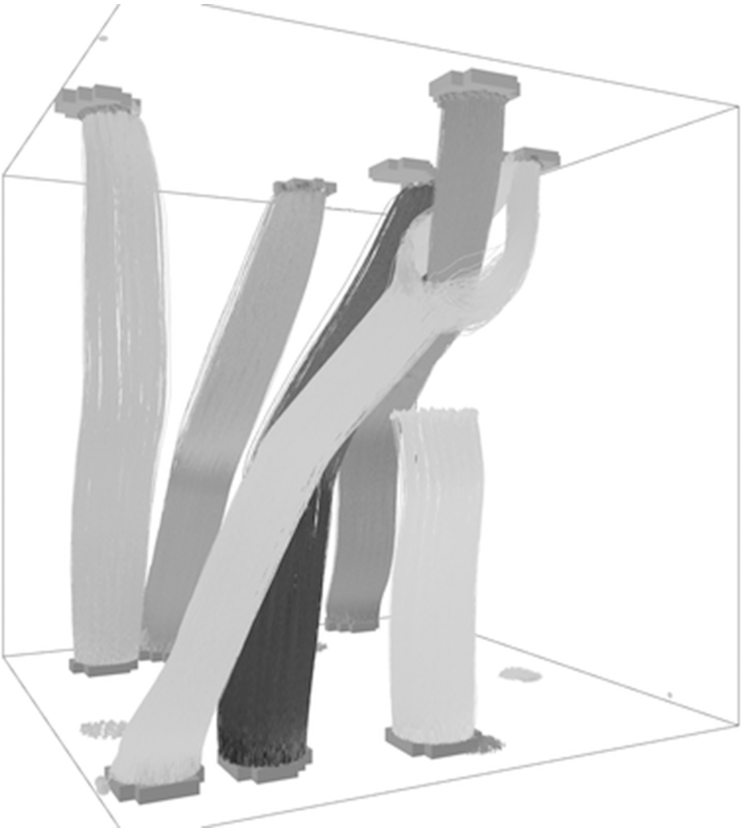
Figure 6 shows a typical simulation of the growth of five fiber bundles, each comprising 5000 fibers. Origins and destinations were randomly chosen on the lower and upper surfaces, respectively. The simulation was modified from a flocking model by Wilenski (2005) implemented in NetLogo 6.0 (Wilenski, 1999). We have observed that sometimes a microrobot gets separated from its cohort, or that a cohort might split into two to get around another fiber bundle (see Figure 6), but the microrobots usually reach their destinations. In this case, out of 25 000 connections, 269 were incorrect, which is approximately 1%. Perhaps due to one bundle having to split so near its destination, this is atypically bad performance; accuracies of 99.9% are more common. The bundles occupy approximately 10% of the volume in this simulation. Figure 7 shows a continuation of the same simulation from a different viewpoint which shows the sixth bundle under construction.

Figure 6. Simulation of the growth of five fiber bundles, each comprising 5000 fibers. Origins on the lower surface and destinations on the upper surface were chosen randomly. One of the bundles has split to find its way around a previous bundle. Approximately 99% of the connections are correct (which is atypically bad; accuracy of 99.9% is common).



We have run these simulations with from five up to 20 000 microrobots, with exactly the same parameters, and so the algorithm scales over four orders of magnitude. Our goal, however, is to scale up to any number, so for the third phase of our algorithm development, we adopt the continuous approximation, treating an indefinitely large microrobot swarm as a continuous mass. We could have the microrobot mass move like a rigid body, with fixed spatial relations between the robots (with a result like Figs. 6 and 7), but such an inflexible approach is not required. Rather, we would like the mass to be able to deform and to split in order to avoid obstacles, as a swarm of discrete agents can do. Therefore, we treat the microrobot mass as a nearly incompressible fluid, so that the mass maintains an approximately constant density of agents no matter how it moves. To accomplish this, we use a

Figure 7. Simulation of the growth of six fiber bundles, each comprising 5000 fibers. This is a continuation of the simulation in Figure 6 (from a different viewpoint) and shows the sixth bundle under construction.



spatially continuous flocking algorithm (cf. Carrillo, Fornasier, Toscani & Vecil, 2010; Chuang, D'Orsogna, Marthaler, Bertozzi & Chayes, 2007; Topaz, Bertozzi & Lewis, 2006).

Rather than using a repellant morphogen to avoid collision, the existing paths clamp the attractant to 0 (e.g., by quickly degrading it); they function as attractant sinks. Since the concentration of attractant decreases exponentially with distance from the goal, a particular rate of repellant release would have a greater effect around obstacles far from the goal compared to those near it, and it would be difficult to find a rate that works well at all distances. On the other hand, having the existing paths clamp the attractant to 0 has the same relative effect independent of attractant concentration, so obstacles are treated the same way regardless of their distance from the goal.

Let A be the attractant concentration, G the goal-region density, and P the existing path density. Then the production, diffusion, decay, and clamping of the attractant is governed by the change equation:

$$\text{Đ}A = D_A \nabla^2 A - A / \tau_A + \kappa_G G(1 - A) - PA / \tau_P.$$

Let C be the density of microrobots. We define a potential function $U(C)$ that is zero for acceptable robot densities and increases rapidly for densities outside this range (either too great or too small). Our goal is for the robot mass to move up the A gradient without the density getting out of bounds, and so the desired velocity $\mathbf{v}'$ is proportional to the difference of the versor of the attractant gradient and the gradient of the density potential:

$$\mathbf{v}' = \frac{\kappa_A \nabla A}{\|\nabla A\|} - \lambda \nabla U(C).$$

where κ_A is the swarm speed and λ governs the importance of constant density. This value is undefined, however, if the gradient is zero, and more realistically, we should ignore the gradient if it is smaller than some specified threshold ϑ_A. Therefore, we alter the velocity equation to ignore a weak gradient:

$$s = \|\nabla A\|,$$
$$V' = [s > \vartheta_A] \kappa_A \nabla A / s - \lambda \nabla U(C).$$

We set the acceleration (in the material frame) to $\mathbf{a} = r(\mathbf{v}' - \mathbf{v})$ which steers the mass in the desired direction at a rate r. Subtracting the convection term gives the change in velocity in the spatial frame, $\mathbf{a} - \mathbf{v} \cdot \nabla \mathbf{v}$ therefore the change in the velocity field in the spatial frame is:

$$\text{Đ}\mathbf{v} = r\left(\left[s > \vartheta_A\right] \kappa_A \nabla A / s - \lambda \nabla U(C) - \mathbf{v}\right) - \mathbf{v} \cdot \nabla \mathbf{v}.$$

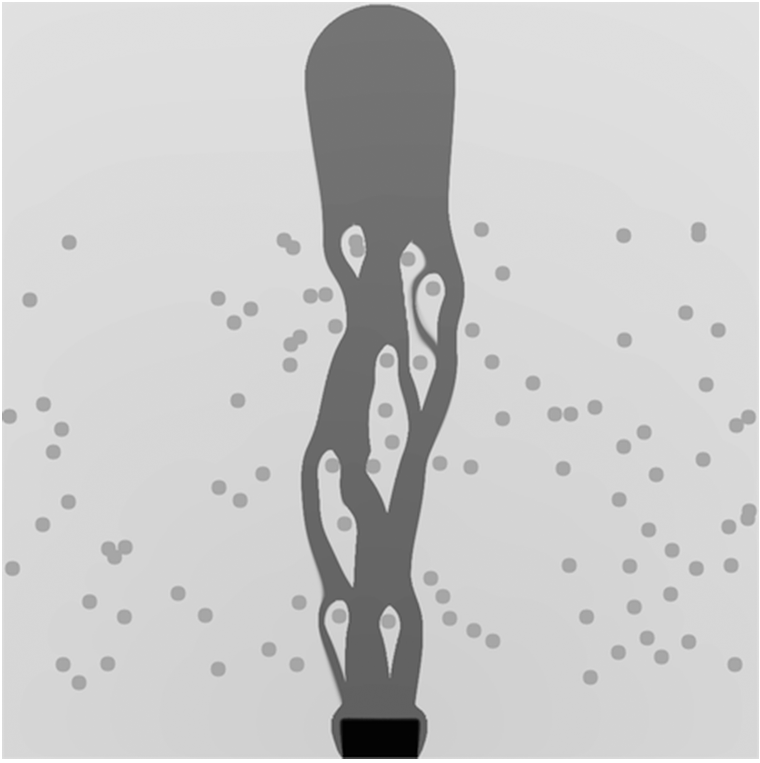
The microrobot flux is $C\mathbf{v}$, and therefore the changes in robot density and path concentration P are governed by:

$$\text{Đ}C = -\nabla \cdot Cv,$$
$$\text{Đ}P = \kappa_p C(1 - P).$$

The equation for $\text{Đ}C$ describes the changing number density of the microrobot mass in the spatial frame.

Figure 8 shows a two-dimensional simulation of the system of PDEs in the Eulerian reference frame. It shows a single path deposited by a continuous mass of microrobots from an origin at the center of the top edge to a destination in the center of the bottom edge, which avoids many obstacles (e.g., previously generated paths). In order to pass between the closely spaced obstacles, the swarm has to split into several streams.

Figure 8. Two-dimensional simulation of path creation by massive robot swarm. The attractant diffuses from the destination at the bottom center. The robot swarm emerges from the origin at the top center and navigates its way between many obstacles representing previously grown paths in cross section. The swarm has to split into multiple streams to pass between the closely spaced obstacles.



FUTURE RESEARCH DIRECTIONS

Many research issues remain in the application of continuum mechanics to the coordination of massive robot swarms in artificial morphogenesis: Is biological morphogenesis the best model for the assembly of complex hierarchical structures by microrobot swarms? Is continuum mechanics a good mathematical framework for describing artificial morphogenetic processes? Are partial differential equations a useful tool for describing the behavior of massive swarms of microrobots? Can we easily program artificial morphogenetic processes to assemble a wide variety of useful structures? Can these programs be easily scaled *down* to large but finite swarms of finite-size microrobots? Can we automatically—or at least systematically—translate morphogenetic programs expressed in PDEs into control processes for massive swarms of physical microrobots?

In support of these research topics, there are several specific directions for this research. First, it is essential to stay engaged with developmental biology. As embryologists unravel morphogenetic processes, we can learn from them and apply the processes to the assembly of the same or similar structures. Even hypotheses that are ultimately disproved in a biological context, may be applied in artificial morphogenesis if they are mathematically sound. Conversely, research in artificial morphogenesis may suggest hypotheses for developmental biology.

Second, we must continue to apply artificial morphogenesis to the simulated assembly of useful structures (such as the examples in this chapter) in order to learn the art of morphogenetic engineering and discover where the techniques can be improved. We have found a useful source of problems and test cases to be the assembly of future robots with sense organs, motor systems, and artificial control systems approaching the complexity of mammalian nervous systems, but there are many other application areas.

While PDEs remain the principal tool for expressing artificial morphogenetic processes, the morphogenetic programming notation should be further refined to facilitate expression and testing of these processes by simulation (*in silico*) and eventually with real microrobots (*in materia*). The notation evolves as we gain experience using it to program swarms of microscopic agents.

Morphogenetic processes have many parameters (e.g., diffusion rates, emission rates, decay rates, velocities, thresholds) that must be adjusted for the processes to succeed. In general, we seek processes that are not overly sensitive to these parameters, and embryology is a guide to robust morphogenesis. Nevertheless, it is often difficult to find regions of the parameter space that work well and we are planning to develop both mathematical guidelines and machine learning techniques to find them more quickly. This is essential for a mature morphogenetic engineering discipline.

Some developmental processes are commonly described in terms of individual cells (such as neurons) and require some reconceptualization to be expressed as PDEs. This might seem to indicate that PDEs are not the best model for morphogenesis in these cases, but this reconceptualization often reveals issues that need to be addressed to achieve scalability. Therefore it is a worthwhile exercise. In particular, we consider the assembly of complex artificial neural systems to be a good test case for artificial morphogenesis.

We are investigating the global-to-local compilation problem, but are confident that our continuum mechanics approach will facilitate the translation from infinitesimal particles to finite-size microrobots. In particular, smoothed particle hydrodynamics looks promising for bridging this gap, but other related techniques are worth investigation. Our PDE-based morphogenetic algorithms assume the existence of idealized infinitesimal particles, and so the behavior of more realistic finite-size microrobots should also be simulated, for example in a physics engine. Our investigation of the global-to-local compilation problem will address this issue as well.

More research is required on the capabilities that may be reasonably expected from future manufactured microrobots as well as from genetically engineered microorganisms serving as microrobots. Important issues are their capabilities for emitting and sensing signals, for moving, for adhering to each other, and for proliferation. The complexity and topology (e.g., analog or digital) of their control mechanisms is also important.

CONCLUSION

We have described an approach to coordinating massive microrobot swarms that takes the number of agents to the continuum limit. This allows the use of partial differential equations and continuum mechanics, which guarantees that our coordination strategies scale up to very large number of microrobots (hundreds of thousands, millions, or more). In contrast, alternative approaches to swarm control often have limited scalability or make assumptions inapplicable to extremely large swarms of microscopic robots. To illustrate the technique, we have presented artificial morphogenesis as a promising approach to the coordination of microrobot swarms to assemble complex hierarchical structures. It is based on biological morphogenetic processes that are known to be effective and are commonly described in terms of partial differential equations. Other approaches to swarm robotics, in contrast, are often limited to the assembly of relatively simple patterns and structures. The continuum mechanics approach to artificial morphogenesis provides a framework for coordination of massive swarms of microscopic robots for assembly of systems structured from the nanoscale up to the macroscale.

ACKNOWLEDGMENT

I am grateful to Allen McBride for implementing the PDE-based model of path generation and for generating images and videos from the simulation. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors, but was enabled by computing equipment supplied by the University of Tennessee, Knoxville. This chapter is an enhanced version of MacLennan (2014) incorporating new results.

REFERENCES

Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight, T. F. Jr, ... Weiss, R. (2000). Amorphous computing. *Communications of the ACM*, *43*(5), 74–82. doi:10.1145/332833.332842

Agarwal, R., Bohner, M., O'Regan, D., & Peterson, A. (2002). Dynamic equations on time scales: A survey. *Journal of Computational and Applied Mathematics*, *141*(1-2), 1–26. doi:10.1016/S0377-0427(01)00432-0

Ahlbrandt, C. D., & Morian, C. (2002). Partial differential equations on time scales. *Journal of Computational and Applied Mathematics*, *141*(1-2), 35–55. doi:10.1016/S0377-0427(01)00434-4

Bandala, A. A., & Dadios, E. P. (2016). Dynamic aggregation method for target enclosure using smoothed particle hydrodynamics technique: An implementation in quadrotor unmanned aerial vehicles (QUAV) swarm. *Journal of Advanced Computational Intelligence*, *20*(1), 84–91. doi:10.20965/jaciii.2016.p0084

Beysens, D. A., Forgacs, G., & Glazier, J. A. (2000). Cell sorting is analogous to phase ordering in fluids. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(17), 9467–9471. doi:10.1073/pnas.97.17.9467 PMID:10944216

Bohner, M., & Peterson, A. (2001). *Dynamic Equations on Time Scales: An Introduction with Applications*. Boston: Birkäuser. doi:10.1007/978-1-4612-0201-1

Bonabeau, E. (1997). From classical models of morphogenesis to agent-based models of pattern formation. *Artificial Life*, *3*(3), 191–211. doi:10.1162/artl.1997.3.3.191 PMID:9385734

Bourgine, P., & Lesne, A. (Eds.). (2011). *Morphogenesis: Origins of Patterns and Shapes*. Berlin: Springer. doi:10.1007/978-3-642-13174-5

Carrillo, J. A., Fornasier, M., Toscani, G., & Vecil, F. (2010). Particle, kinetic, and hydrodynamic models of swarming. In G. Naldi, L. Pareschi, & G. Toscani (Eds.), *Mathematical Modeling of Collective Behavior in Socio-economic and Life Sciences* (pp. 297–336). Boston: Birkhäuser. doi:10.1007/978-0-8176-4946-3_12

Chuang, Y.-L., D'Orsogna, M. R., Marthaler, D., Bertozzi, A. L., & Chayes, L. S. (2007). State transitions and the continuum limit for a 2D interacting, self-propelled particle system. *Physica D. Nonlinear Phenomena*, *232*(1), 33–47. doi:10.1016/j.physd.2007.05.007

Cickovski, T. M., Huang, C., Chaturvedi, R., Glimm, T., Hentschel, H. G. E., Alber, M. S., ... Izaguirre, J. A. (2005). A framework for three-dimensional simulation of morphogenesis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *2*(4), 273–278. doi:10.1109/TCBB.2005.46 PMID:17044166

Cooke, J., & Zeeman, E. C. (1976). A clock and wavefront model for control of the number of repeated structures during animal morphogenesis. *Journal of Theoretical Biology*, *58*(2), 455–476. doi:10.1016/S0022-5193(76)80131-2 PMID:940335

Correll, N. (2007). *Coordination Schemes for Distributed Boundary Coverage with a Swarm of Miniature Robots: Synthesis, Analysis and Experimental Validation* (PhD thesis). Ecole Polytechnique Fédérale de Lausanne.

Correll, N., & Martinoli, A. (2006). System identification of self-organizing robotic swarms. In M. Gini, & R. Voyles (Eds.), *Proceedings of the 8th International Symposium on Distributed Autonomous Robotic Systems (DARS 2006)* (pp. 31–40). Heidelberg, Germany: Springer.

de Gennes, P. G. (1992). Soft matter. *Science*, *256*(5056), 495–497. doi:10.1126cience.256.5056.495 PMID:17787946

Dequéant, M. L., & Pourquié, O. (2008). Segmental patterning of the vertebrate embryonic axis. *Nature Reviews. Genetics*, *9*(5), 370–382. doi:10.1038/nrg2320 PMID:18414404

Doursat, R. (2008). Organically grown architectures: Creating decentralized, autonomous systems by embryomorphic engineering. In R. P. Würtz (Ed.), *Organic Computing* (pp. 167–200). New York: Springer.

Doursat, R., Sayama, H., & Michel, O. (Eds.). (2012). *Morphogenetic Engineering: Toward Programmable Complex Systems*. New York: Springer. doi:10.1007/978-3-642-33902-8

Edelman, G. M. (1988). *Topobiology: An Introduction to Molecular Embryology*. New York: Basic Books.

Feddema, J. T., Lewis, C., & Schoenwald, D. A. (2002). Decentralized control of cooperative robotic vehicles: Theory and application. *IEEE Transactions on Robotics and Automation*, *18*(5), 852–864. doi:10.1109/TRA.2002.803466

Fleischer, K. W. (1995). A multiple-mechanism developmental model for defining self-organizing geometric structures (Doctoral dissertation, California Institute of Technology, 1995). *Dissertation Abstracts International B*, 56/09, 4981, 1996.

Forgacs, G., & Newman, S. A. (2005). *Biological Physics of the Developing Embryo*. Cambridge, UK: Cambridge University Press. doi:10.1017/CBO9780511755576

Gazi, V., & Passino, K. M. (2003). Stability analysis of swarms. *IEEE Transactions on Automatic Control*, *48*(4), 692–697. doi:10.1109/TAC.2003.809765

Gingold, R. A., & Monaghan, J. J. (1977). Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, *181*(3), 375–389. doi:10.1093/mnras/181.3.375

Goldstein, S. C., Campbell, J. D., & Mowry, T. C. (2005). Programmable matter. *Computer*, *38*(6), 99–101. doi:10.1109/MC.2005.198

Hamann, H. (2010). *Space-time Continuous Models of Swarm Robotic Systems: Supporting Global-to-Local Programming*. Berlin: Springer. doi:10.1007/978-3-642-13377-0

Helbing, D., Schweitzer, F., Keltsch, J., & Molnar, P. (1997). Active walker model for the formation of human and animal trail systems. *Physical Review. E*, *56*(3), 2527–2539. doi:10.1103/PhysRevE.56.2527

Hogg, T. (2006). Coordinating microscopic robots in viscous fluids. *Autonomous Agents and Multi-Agent Systems*, *14*(3), 271–305. doi:10.100710458-006-9004-3

Kitano, H. (1996). Morphogenesis for evolvable systems. In E. Sanchez & M. Tomassini (Eds.), *Towards Evolvable Hardware: The Evolutionary Engineering Approach* (pp. 99–117). Berlin: Springer. doi:10.1007/3-540-61093-6_5

Kornienko, S., Kornienko, O., & Levi, P. (2005). Swarm embodiment – a new way for deriving emergent behavior in artificial swarms. In P. Levi, M. Schanz, R. Lafrenz, & V. Avrutin (Eds.), *Autonome Mobile Systeme* (pp. 25–32). Berlin: Springer.

Lerman, K. (2004). A model of adaptation in collaborative multi-agent systems. *Adaptive Behavior*, *12*(3–4), 187–198. doi:10.1177/105971230401200305

Lerman, K., & Galstyan, A. (2002). Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, *13*(2), 127–141. doi:10.1023/A:1019633424543

Lerman, K., Martinoli, A., & Galstyan, A. (2005). A review of probabilistic macroscopic models for swarm robotic systems. In E. Şahin & W. M. Spears (Eds.), Swarm Robotics 2004 (pp. 143–152). Heidelberg, Germany: Springer. doi:10.1007/978-3-540-30552-1_12

Liu, G. R., & Liu, M. B. (2003). *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. Singapore: World Scientific. doi:10.1142/5340

MacLennan, B. J. (2010). Morphogenesis as a model for nano communication. *Nano Communication Networks*, *1*(3), 199–208. doi:10.1016/j.nancom.2010.09.007

MacLennan, B. J. (2011). Artificial morphogenesis as an example of embodied computation. *International Journal of Unconventional Computing*, *7*(1–2), 3–23.

MacLennan, B. J. (2012a). Embodied computation: Applying the physics of computation to artificial morphogenesis. *Parallel Processing Letters*, *22*(3), 1240013. doi:10.1142/S0129626412400130

MacLennan, B. J. (2012b). Molecular coordination of hierarchical self-assembly. *Nano Communication Networks*, *3*(2), 116–128. doi:10.1016/j.nancom.2012.01.004

MacLennan, B. J. (2014). Coordinating massive robot swarms. *International Journal of Robotics Applications and Technologies*, *2*(2), 1–19. doi:10.4018/IJRAT.2014070101

MacLennan, B. J. (in press). Coordinating swarms of microscopic agents to assemble complex structures. In Y. Tan (Ed.), *Swarm Intelligence — From Concepts to Applications (PBCE110)*. London: IET.

Martinoli, A. (1999). *Swarm Intelligence in Autonomous Collective Robotics: From Tools to the Analysis and Synthesis of Distributed Control Strategies* (PhD thesis). Ecole Polytechnique Fédérale de Lausanne.

Martinoli, A., Easton, K., & Agassounon, W. (2004). Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *The International Journal of Robotics Research*, *23*(4), 415–436. doi:10.1177/0278364904042197

Meinhardt, H. (1982). *Models of Biological Pattern Formation*. London: Academic Press.

Meng, Y., & Jin, Y. (Eds.). (2011). *Bio-inspired Self-organizing Robotic Systems. In Studies in Computational Intelligence* (Vol. 355). Berlin: Springer. doi:10.1007/978-3-642-20760-0

Murata, S., & Kurokawa, H. (2007). Self-reconfigurable robots: Shape-changing cellular robots can exceed conventional robot flexibility. *IEEE Robotics & Automation Magazine*, *14*(1), 71–78. doi:10.1109/MRA.2007.339607

Nagpal, R., Kondacs, A., & Chang, C. (2003). Programming methodology for biologically-inspired self-assembling systems. *AAAI Spring Symposium on Computational Synthesis: From Basic Building Blocks to High Level Functionality*. Retrieved from http://www.aaai.org/Papers/Symposia/Spring/2003/SS-03-02/SS03-02-024.pdf

Nüsslein-Volhard, C. (2008). *Coming to Life: How Genes Drive Development*. Carlsbad, CA: Kales.

Pac, M. R., Erkmen, A. M., & Erkmen, I. (2007). Control of Robotic Swarm Behaviors Based on Smoothed Particle Hydrodynamics. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4194–4200). IEEE Press. 10.1109/IROS.2007.4399437

Perkinson, J. R., & Shafai, B. (2005). A decentralized control algorithm for scalable robotic swarms based on mesh-free particle hydrodynamics. In *Proceedings of the IASTED International Conference on Robotics and Applications* (pp. 102–107). Cambridge, MA: IASTED.

Pimenta, L. C. A., Mendes, M. L., Mesquita, R. C., & Pereira, G. A. S. (2007). Fluids in electrostatic fields: An analogy for multirobot control. *IEEE Transactions on Magnetics*, *43*(4), 1765–1768. doi:10.1109/TMAG.2007.892514

Pimenta, L. C. A., Pereira, G. A. S., Michael, N., Mesquita, R. C., Bosque, M. M., Chaimowicz, L., & Kumar, V. (2013). Swarm coordination based on smoothed particle hydrodynamics technique. *IEEE Transactions on Robotics*, *29*(2), 383–399. doi:10.1109/TRO.2012.2234294

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *Computer Graphics*, *21*(4), 25–34. doi:10.1145/37402.37406

Salazar-Ciudad, I., Jernvall, J., & Newman, S. (2003). Mechanisms of pattern formation in development and evolution. *Development*, *130*(10), 2027–2037. doi:10.1242/dev.00425 PMID:12668618

Schweitzer, F. (2002). Brownian agent models for swarm and chemotactic interaction. In D. Polani, J. Kim, T. Martinetz (Eds.), *Fifth German Workshop on Artificial Life: Abstracting and Synthesizing the Principles of Living Systems* (pp. 181–190). Berlin: Akademische Verlagsgesellschaft Aka.

Schweitzer, F. (2003). Brownian agents and active particles. In On the Emergence of Complex Behavior in the Natural and Social Sciences (pp. 295–333). Berlin: Springer.

Schweitzer, F., Lao, K., & Family, F. (1997). Active random walkers simulate trunk trail formation by ants. *Bio Systems*, *41*(3), 153–166. doi:10.1016/S0303-2647(96)01670-X PMID:9113350

Song, Z., Lipinski, D., & Mohseni, K. (2017). Multi-vehicle cooperation and nearly fuel-optimal flock guidance in strong background flows. *Ocean Engineering*, *141*, 388–404. doi:10.1016/j.oceaneng.2017.06.024

Spector, L., Klein, J., Perry, C., & Feinstein, M. (2005). Emergence of collective behavior in evolving populations of flying agents. *Genetic Programming and Evolvable Machines*, *6*(1), 111–125. doi:10.100710710-005-7620-3

Spicher, A., Michel, O., & Giavitto, J. (2005) Algorithmic self-assembly by accretion and by carving in MGS. In *Proceedings of the 7th International Conference on Artificial Evolution (EA '05)* (pp. 189–200). Berlin: Springer-Verlag.

Taber, L. A. (2004). *Nonlinear Theory of Elasticity: Applications in Biomechanics*. Singapore: World Scientific. doi:10.1142/5452

Teo, J. J. Y., Woo, S. S., & Sarpeshkar, R. (2015). Synthetic biology: A unifying view and review using analog circuits. *IEEE Transactions on Biomedical Circuits and Systems*, *9*(4), 453–474. doi:10.1109/TBCAS.2015.2461446 PMID:26372648

Topaz, C. M., Bertozzi, A. L., & Lewis, M. A. (2006). A nonlocal continuum model for biological aggregation. *Bulletin of Mathematical Biology*, *68*(7), 1601–1623. doi:10.100711538-006-9088-6 PMID:16858662

Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *237*(641), 37–72. doi:10.1098/rstb.1952.0012

Vicsek, T., Czirok, A., Ben-Jacob, E., Cohen, I., & Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, *6*(75), 1226–1229. doi:10.1103/PhysRevLett.75.1226 PMID:10060237

Wilensky, U. (1999). *NetLogo*. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University; http://ccl.northwestern.edu/netlogo/

Wilensky, U. (2005). *NetLogo Flocking 3D Alternate model*. Retrieved from http://ccl.northwestern.edu/netlogo/models/

Winfield, A. F. T., Sav, J., Fernández-Gago, M.-C., Dixon, C., & Fisher, M. (2005). On formal specification of emergent behaviours in swarm robotic systems. *International Journal of Advanced Robotic Systems*, *2*(4), 363–370. doi:10.5772/5769

Yamins, D. (2005). Towards a theory of "local to global" in distributed multi-agent systems. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMS 2005)* (pp. 183–190). Utrecht, The Netherlands: Academic Press. 10.1145/1082473.1082501

Yamins, D. (2007). *A Theory of Local-to-Global Algorithms for One-Dimensional Spatial Multi-Agent Systems* (PhD thesis). Cambridge, MA: Harvard University.

Yamins, D., & Nagpal, R. (2008). Automated global-to-local programming in 1-D spatial multi-agent systems. In L. Padgham, D. C. Parkes, J. P. Müller, S. Parsons (Eds.), *Proceedings of 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)* (pp. 615–622). Estoril, Portugal: Academic Press.

KEY TERMS AND DEFINITIONS

Artificial Morphogenesis: A process for assembling or generating physical structures modeled more or less closely on biological *morphogenesis* (q.v.).

Body: In the context of *artificial morphogenesis* (q.v.), a specific instance of a *substance* (q.v.) occupying a region of space, which may vary as determined by the behavior of its constituent particles. A massive swarm of microrobots is treated as a body.

Caudal: Referring to the tail or posterior part of a structure.

Change Equation: An equation expressing the change in a quantity over time that can be interpreted either as a differential equation in continuous time or as a difference equation in discrete time. The change operator is written $Đ$ (uppercase *eth*).

Eulerian Frame of Reference: An approach to continuum mechanics in which spatially distributed variables are associated with fixed locations in space. Also called *spatial frame of reference*.

Lagrangian Frame of Reference: An approach to continuum mechanics in which spatially distributed variables are associated with fixed particles in a fluid or other substance, which may be moving through space. Also called *material frame of reference*.

Material Frame of Reference: Synonymous with *Lagrangian frame of reference* (q.v.).

Morphogen: A substance that diffuses and governs *morphogenesis* (q.v.) by means of variation in concentration.

Morphogenesis: The development of three-dimensional physical form during the development of an embryo.

Rostral: Referring to the head or anterior part of a structure.

Spatial Frame of Reference: Synonymous with *Eulerian frame of reference* (q.v.).

Substance: In the context of *artificial morphogenesis* (q.v.), a class of infinitesimal particles with particular defined properties and behaviors (analogous to a class in object-oriented programming). Substances are used to define the behavior of particular kinds of microrobots. We distinguish *physical substances*, which represent the relatively fixed physical properties of agents (e.g., mass, size, sensors, actuators), from *controllable substances*, which represent the programmable aspects of agent behavior.

Chapter 5

Improving Dependability of Robotics Systems:

Analysis of Sequence-Dependent Failures

Nidhal Mahmud
SYSAF, UK

ABSTRACT

In this chapter, the authors propose an algorithm for the reduction of fault tree expressions that are generated from failure behavioral models. The significance of the sequencing of events is preserved during the generation and all along the reduction process, thus allowing full qualitative analysis. Thorough and detailed analysis results should positively impact the design of condition monitoring and failure prevention mechanisms. A behavioral model of a robotic system that exhibits sequence-dependent failures is used in the study.

INTRODUCTION

The use of robotics systems is widespread and spans a variety of application areas. From healthcare, to manufacturing, to nuclear power plants, to space missions, these systems are typically conceived to perform difficult, dangerous or critical tasks. The nature of such tasks—e.g., surgery operations, radioactive waste clean-up or space mining—places high demands on the dependability (reliability, safety, availability, and maintainability) of robotics systems.

DOI: 10.4018/978-1-5225-5276-5.ch005

The preoccupations in the dependability of robotics systems are not new. Fault Tree Analysis (FTA, Vesely 1981) and Failure Modes and Effects Analysis (FMEA, IEEE Std.352 1987) are among the most often used techniques in various domains of robotics. For instance, Visinsky, Walker, and Cavallaro (1993) describe the use of FTA for robots operating in remote and hazardous environments. Other fields of application include industrial robots like in (Karbasian, Mehr, & Agharajabi, 2012), and modular and swarm robots like in (Murray, Liu, Winfield, Timmis, & Tyrrell, 2012).

The widespread use of FTA in the dependability assessment of complex systems is mainly due to the flexibility and ease of use of the fault trees. These are static (i.e., 'pure' Boolean) models, and therefore enable the use of efficient Boolean calculus in the elimination of component failures that are irrelevant to the total failure of the system. This logical reduction (known as qualitative analysis) simplifies the process to produce overall probabilities of system hazards (i.e., quantitative analysis). Nevertheless, such convenience comes with the loss of the significance of the sequencing of failure events—i.e., the dynamic features often exhibited by modern systems cannot be captured by combinatorial models like this type of fault trees.

Robotics systems are certainly not an exception when it comes to sequence-dependent failures. For example, preclusion of the dynamic aspects due to the use of static fault trees in the analysis of modular robotic systems is clearly noted in (Murray et al., 2012). To overcome such drawback, an alternative can be the utilization of fault trees that are extended with capabilities to capture the dynamic features. A well-known example is the Dynamic Fault Tree (DFT) approach (Dugan, Bavuso, & Boyd, 1992). This method was primarily conceived for quantitative analysis, which is often state-based—i.e., Markov analysis which is based on state transition diagrams (Markov models) is the DFT most prominent solving technique. That is, the full power of the Boolean methods was sacrificed here, especially when it comes to analyzing the dynamic parts of the system at the level of the fault tree (i.e., reducing the DFT).

Theoretically, some later research efforts have provided workarounds to the question of FTA with dynamic aspects. To deal with it, a technique which is relevant to this chapter consists of extending the Boolean methods with temporal logic calculus. In this connection, a set of temporal laws that enable qualitative analysis of fault trees extended with dynamic features can be found in (Walker & Papadopoulos, 2009). In the same vein, the algebraic formalism in (Merle, Roussel, Lesage, & Bobbio, 2010) proposes formal descriptions of dynamic behaviors and provides proofs of a number of theorems useful for the qualitative analysis of this type of fault trees. The latter approach also deals with the corresponding probabilistic algebraic analysis.

In practice, automation of such advanced FTA as part of integrated dependability and systems engineering processes requires an automated generation and synthesis of these fault trees from failure behavioral models that are linked to the system specifications. The work in (Mahmud, Walker, & Papadopoulos, 2012) describes a suitable approach to generating and synthesizing fault trees that preserve the significance of the event-order from hierarchical models. Application areas for this approach include the automotive domain (Chen, Mahmud, Walker, Feng, Lönn, & Papadopoulos, 2013) and real-time performance-critical distributed computer systems in general (Mahmud, & Mian, 2013; Mian, Bottaci, Papadopoulos, Sharvia, & Mahmud, 2015). More details about integration in an extended FTA through a Model-Based development process can be found in (Kolagari, Chen, Lanusse, Librino, Lönn, Mahmud, Mraidha, Reiser, Torchiaro, Tucci-Piergiovanni, Wägemann, & Yakymets, 2015).

In this chapter, emphasis is put on the significance of the sequencing of failure events and its implications in dependability analysis of robotics systems using FTA. Accordingly, a suitable technique for an automated generation and analysis of extended fault trees from system models is presented. Furthermore, we outline a novel approach to automated reduction of these fault tree expressions. The chapter is structured as follows: the background section provides a literature review and highlights the relevant approaches to generating fault trees from systems models for analysis. The next section outlines an algorithm for the logical reduction of fault tree algebraic expressions that are extended with temporal semantics. Finally, we present some future research directions, then we conclude.

BACKGROUND

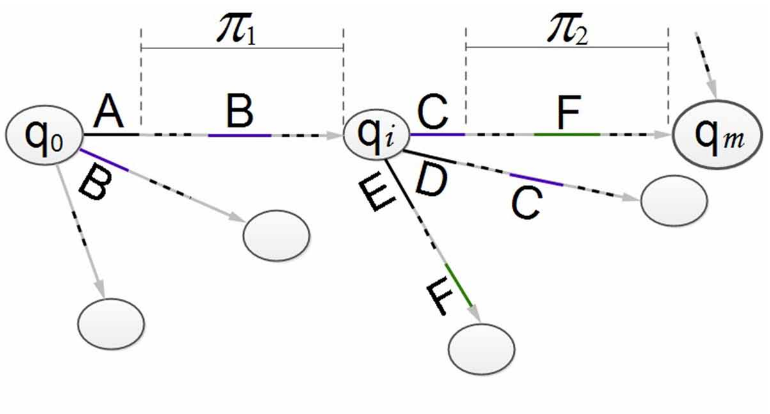
A number of approaches to generating fault trees from system models can be found in the literature. However, a special focus is put on automata representations of the dysfunctional behaviors of systems from which the fault trees get generated. In this regard, the existing approaches fall into two main categories. The first category concerns the generation of static fault trees; it includes approaches that have been used in the context of influential modeling languages like Altarica (Rauzy, 2002) and AADL (Joshi, Vestal, & Binns, 2007). The former has been used in several aerospace projects including Airbus civil aircraft programs. The latter is increasingly being accepted by the aerospace community as a future standard. Moreover, there have been efforts recently to generate static fault trees from modeling languages specific to robotics systems, e.g. RobotML in (Yakymets, Dhouib, Jaber, & Lanusse, 2013).

The second category of methods for fault tree generation includes approaches to generating dynamic fault trees (Dehlinger, & Dugan, 2008), and fault trees that are extended with the significant temporal information about the order in which failure events occur (Mahmud, Papadopoulos, & Walker, 2010; Mahmud et al., 2012;

Mahmud, & Mian, 2013). The DFTs are solved quantitatively, often, by converting them into equivalent Markov models. Although, there have been efforts on the minimization of the obtained Markov models (Crouzen, Hermanns, & Zhang, 2008; Boudali, Crouzen, & Stoelinga, 2010), there is still less focus on qualitative analysis at the level of the fault tree in this method. However, the approach in (Mahmud et al., 2010, 2012; Mahmud, & Mian, 2013) allows to reduce the extended fault trees into equivalents which can be proven by using the temporal laws and temporal truth tables in (Walker & Papadopoulos, 2009; Merle et al., 2010). This is a State Automata-based top-down deductive approach to Fault tree synthesis with Order-dependent behaviors, see SAFORA in (Mahmud, 2012). It has demonstrated its value in the framework of a rich model-based design founded on EAST-ADL (Chen et al., 2013; Kolagari et al., 2015). EAST-ADL is an approach for describing automotive electronic systems through an information model that captures engineering information in a standardized form (EAST-ADL, 2010).

The fault tree generation algorithm used in the Safora method is based on the approach outlined in (Mahmud et al., 2010). It converts State Machines (SMs) showing transitions from normal to degraded and failed states into fault tree algebraic expressions. The transitions are labeled by failure events (denoted by upper case letters taken from the alphabet, e.g. A, B, C … like in Figure 1) and the states that are relevant to the system dependability are marked (denoted by the lowercase letter q with the subscript m). An algebraic expression is a logical combination of failure events —using the logical AND (symbol '.') and the logical OR (symbol '+') — which can be extended with sequence information by using the Priority-OR

Figure 1. Example of a SM path for the fault tree algebraic representation



$$q_m = A|B.\pi_1.C|D|E.\pi_2 + \ldots$$

temporal logic operator (symbol '|'). The latter, abridged as POR, can be found in (Walker & Papadopoulos, 2009) and represents a priority situation where one event must occur first and other events may or may not occur subsequently. For example, in the expression 'A|B', A occurs and B does not occur; or A and B both occur, but in sequence. Similarly, in A|B|C, A occurs first or alone; should B and/or C occur, this must happen after A, but the order in which B and C occur is unimportant.

A sequence can be represented by using only '.' and '|', which is a central issue in (Mahmud et al., 2010). For example, the sequence A occurs first then B occurs second, which is denoted by A<B ('<' represents the Priority-AND operator, which is abridged as PAND), can be represented by A|B.B, where '|' has priority over '.'. Proofs of this equivalence and others that are useful to this chapter, like the following, can be found in (Walker & Papadopoulos, 2009; Merle et al., 2010). We wish to note, though, that POR is referred to as non-inclusive BEFORE in (Merle et al., 2010).

$$A<B \Leftrightarrow (A|B).B$$

$$A<B + B<A \Leftrightarrow A.B$$

$$(A|B)\ .\ (A|C) \Leftrightarrow A|B|C$$

$$A|(B+C) \Leftrightarrow A|B|C$$

For the sake of clarity, Table 1 represents a temporal truth table that demonstrates that A<B and (A|B).B are equivalent. Such tables are analogous to the Boolean truth tables but take into account the temporal order in which events occur. The value 0 in a cell means that the corresponding event has not occurred. Contrariwise, any non-zero value means that the event has occurred, possibly before (resp. after) any other event in the same row with a non-zero value that is greater (resp. smaller). For example, in the second to last row, A occurs first (value 1) and B occurs second (value 2). In the last row, the inverse is observed regarding the temporal order in which A

Table 1. A temporal truth table

A	B	A<B	A|B	(A|B).B
0	0	0	0	0
0	1	0	0	0
1	0	0	1	0
1	2	2	1	2
2	1	0	0	0

and B occur. We wish to note, though, that the table does not show a row in which A and B have the same non-zero value (in which case A<B, A|B and thus (A|B).B would each evaluate to 0). That is, we assume that simultaneity of occurrences of events is due to common cause failures, e.g., a power failure which causes several components to fail at the same time. Such failures are treated separately and are assigned separate failure rates.

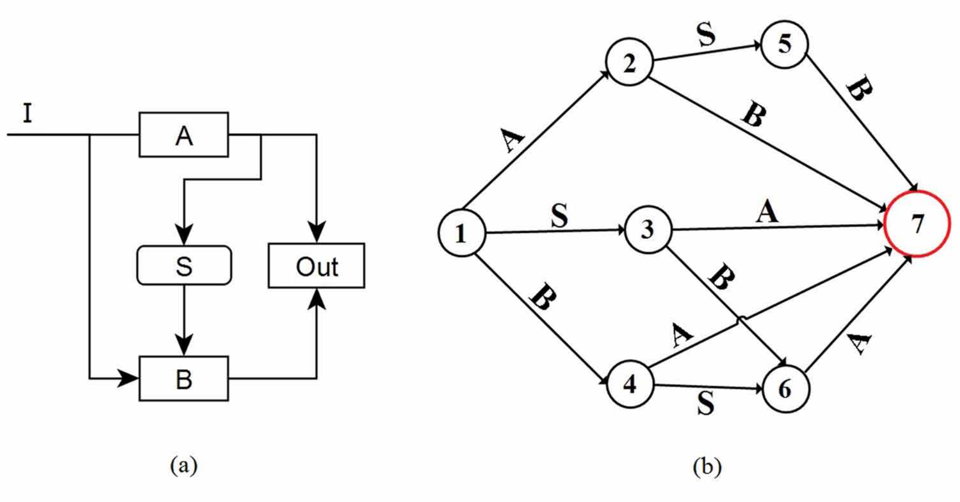
In (Mahmud et al., 2010), the algebraic expression deduced from a marked state (denoted by q_m) in a SM is essentially the disjunction over the paths π from q_m to the initial state (traversed backwards) of the conjunction of the events that label π. Moreover, an event E, which is incident from a state q in which paths diverge, will have priority over any other event E', also incident from q, if and only if there is an event in the path starting from q through E to q_m, which is repeated (i.e., reappearing) in a path starting from q through E' to any state. For example, the (sub) path from q_i to q_m (top right of Figure 1) is essentially represented by $C.\pi_2$, where C is the sub-path's event which is incident from q_i and π_2 is the conjunction of the remaining events to reach q_m. Since C is repeated in a path divergent from q_i (slightly downwards in the figure), also there is an event F in π_2 which is repeated in another divergent path (to the bottom of the figure), C will then have priority over D and E, the other events which are also incident from q_i. Therefore, $C.\pi_2$ is now fine-tuned as $C|D|E.\pi_2$. Similarly, $A.\pi_1$, to the left of Figure 1, is essentially the algebraic representation of the sub path from q_0 (i.e., the initial state) to q_i. This representation is also fine-tuned into $A|B.\pi_1$ since there is an event B that is repeated in a divergent path. Therefore, the full path from q_0 to q_m is algebraically represented by $A|B.\pi_1.\ C|D|E.\pi_2$, and without any loss of the significance of the event order. There might be other paths reaching q_m, which are denoted as disjunctions (sums) of conjunction (product) terms. Thus, the marked state expression is directly obtained in a standard sum-of-product canonical form like in equation (1), which is convenient for the logical manipulation thereof.

$$q_m = \sum\left(\prod E_i \cdot \prod\left(E_j \mid E_k\right)\right),\ j \neq k \tag{1}$$

where '$\sum$' and '$\prod$' represent respectively the Boolean 'OR' and 'AND'.

For example, let us consider the primary-standby (PS) redundant system depicted in Figure 2 (a). "A" is the primary component, "B" is the backup, and "S" is a monitoring sensor whose role is to activate B upon detection of a deviation from the output of A (e.g., omission of output). "I" represents the input to each of the two redundant components, and "Out" is the system output. Out must receive input from either A or B for a continuous operation of the system. This pattern is not uncommon in dependability-critical systems as A and B can be any sensing, control

Figure 2. (A) A Primary-Standby example system and (b) the corresponding failure behavior



or actuating device. A SM example which describes the PS failure behavior is shown on Figure 2 (b), where X∈{A, B, S} denotes an internal failure (i.e., basic event) of the corresponding component to the left of the figure. For the sake of clarity, the SM is simplified by assuming that input I is always provided, Out has no failure mode on its own, and total failure states are all merged into one marked state (i.e., state 7). Figure 2 (b) shows that failure of either primary or backup component needs to be followed, whether immediately or not, by failure of the other unit in order to reach state 7. Also, failure of the sensor is relevant to the total failure of the system only if it occurs too soon—i.e., S fails before failure of the monitored (primary) component occurs. Otherwise, the system can still function in standby mode upon failure of A.

Application of the conversion algorithm outlined previously gives the following algebraic expression for state 7, and in the form of a sum of Product Terms (PTs) as in equation (1):

state 7 =

B.S|B.A|S|B + (PT#1)

A.B|A.S|B|A + (PT#2)

A.S|A.B|S|A + (PT#3)

A|B.S|B|A + (PT#4)

B|S.A|S|B + (PT#5)

A|S.B|S|A (PT#6)

Notice that there are 6 product term expressions, each corresponds to one path reaching state 7.

In fact, each product term $\mathrm{PT} = \prod \mathrm{E_i} \bullet \prod \left(\mathrm{E_j}|\mathrm{E_k}\right)$, $\mathrm{j} \neq \mathrm{k}$ in equation (1) is a cut-sequence set which is also, algebraically, a cut-sequence sum (CSS). For example, A.B|C = {ABC, BAC, BCA, AB, BA}, and where C does not occur in the last 2-event sequences (i.e., AB and BA). Likewise, A.B|C is algebraically the disjunction over all previous sequences. In other words, A.B|C is the set (sum) of all sequences of A, B, and C should C occur, subject to inclusion of any order in which A and B occur (logical AND) and exclusion of the order in which C occurs before B (due to the term 'B|C').

Concerning the expression of state 7, we have: PT#1 is the singleton {ASB}. Similarly, PT#2 = {SBA} and PT#3 = {BSA}. However, PT#4 = {SA, SAB}, PT#5 = {AB, ABS}, and PT#6 = {BA, BAS}. Since PTs can be interpreted as sets or as algebraic expressions, they can be manipulated by unions or by disjunctions (sums). Let's say we have n product terms denoted by $\mathrm{PT_i}$ i=1,2,…,n. The canonical form minimization algorithm proposed in (Merle et al., 2010) removes the redundant elements from the set *S* of all product terms.

$$S = \bigcup_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{PT_i} \text{ (which algebraically is } \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{PT_i}\text{)} \tag{2}$$

According to (Merle et al., 2010), a product term $\mathrm{PT_i}$ is redundant if:

$$\mathrm{PT_i} \,.\, \sum_{\mathrm{j \neq i}} \mathrm{PT_j} = \mathrm{PT_i} \tag{3}$$

However, this algorithm aims at removing the completely redundant PTs. In many cases, we also need to check whether some events are irrelevant, especially when it comes to the order-sensitive part (i.e., the POR input events) of a PT. For instance, while the algorithm doesn't remove any $\mathrm{PT_i}$ i=1,…,6 that compose the canonical form of state 7 of the PS example system, this doesn't mean that they cannot be reduced. Observing, e.g., PT#4 = {SA, SAB}, it shows that B may not occur, or may occur but after A. The other options to reach state 7 are via PT#2 = {SBA}, which means that B occurs between S and A, and via PT#3 = {BSA} which means that B, S, and A, all occur in sequence. In other words, there is a sub-sequence SA common to PT#2, PT#3, and PT#4; whether B occurs before SA, in the middle of SA, after SA, or doesn't occur at all, simply means that B is irrelevant and can be removed, thereby merging these product terms into a reduced (minimal) product term represented by PT#2,3,4 = {SA}. But, the intermediate PT_i i∈{2, 3, 4} need not be removed yet, as they can serve to merge and reduce other PTs.

For example, PT#6 = {BA, BAS} can be merged with PT#2 and PT#3, then reduced into PT#2,3,6 = {BA} —since, overall, whether S is a prefix with BA (i.e., PT#2), S occurs in the middle of BA (i.e., PT#3), or S is a suffix with BA or does not occur at all (i.e., PT#6), all show that S is irrelevant. Similarly, the PT#i i∈{2, 3, 6}need not be removed yet until all iterations are processed. The last iteration concerns PT#5 = {AB, ABS} which can be similarly merged with PT#1 and PT#4, then reduced into PT#1,4,5 = {AB}.

At this stage, state 7 (Figure 2b) canonical form is reduced to S_{min} = PT#2,3,4 ∪ PT#2,3,6 ∪ PT#1,4,5 ∪ $\bigcup_{i=1}^{6} \mathrm{PT\#i}$. That is, S_{min} = {SA, BA, AB}∪ $\bigcup_{i=1}^{6} \mathrm{PT\#i}$ and which is algebraically written as:

$$S_{\mathrm{min}} = \mathrm{A.S|A + A.B|A + B.A|B} + \sum_{i=1}^{6} \mathrm{PT\#i}$$

The intermediate PT#i 1≤i≤6 are now each completely redundant in S_{min} (i.e., $\forall \mathrm{i}\ 1 \leq \mathrm{i} \leq 6\ \mathrm{PT\#i} \bullet \sum_{\substack{\mathrm{PT} \neq \mathrm{PT\#i} \\ \mathrm{PT} \in S_{\mathrm{min}}}} \mathrm{PT} = \mathrm{PT\#i}$) and can all be removed as they cannot serve to reduce a PT any further.

Therefore, S_{min} is now minimized and algebraically written as:

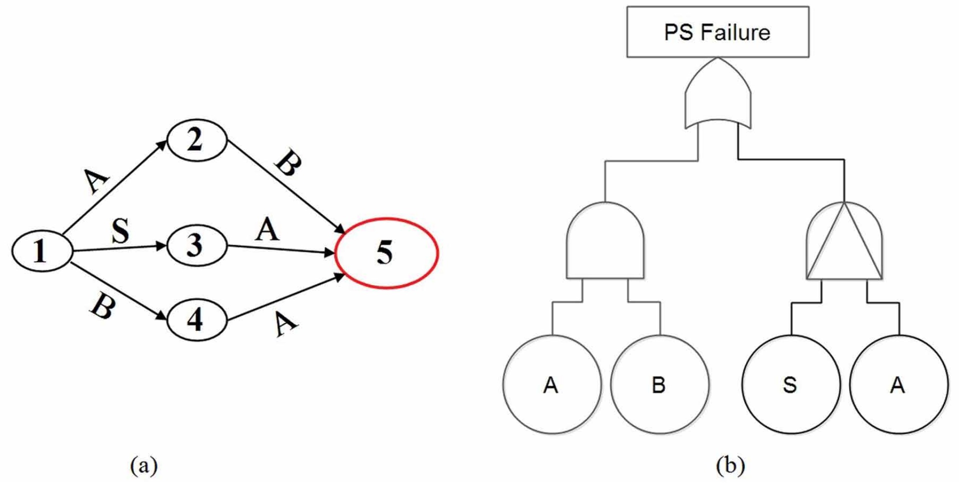
$$S_{\mathrm{min}} = \mathrm{A.S|A + A.B|A + B.A|B} \tag{4}$$

which can also be written as S_{min} = S<A + A.B

It is graphically represented by the reduced SM in Figure 3a (with 3 full paths only instead of 6 initially), and by the reduced fault tree shown in Figure 3b where the gate to the top of the figure is a logical OR, to the left is a logical AND, and to the right is a Priority-AND.

In the following sections, the full reduction algorithm is presented and demonstrated by using a larger example—i.e., a robot joint sensing system with

Figure 3. (A) Reduced SM and (b) Reduced DFT of the PS example system



active, passive and shared redundancies. The example system is relatively small-sized, but exhibits complex dependencies among its components and helps to show the various redundancies which are all removed by the algorithm—i.e., including entirely redundant PTs as well as redundant cut-sequences of PTs.

REDUCTION ALGORITHM

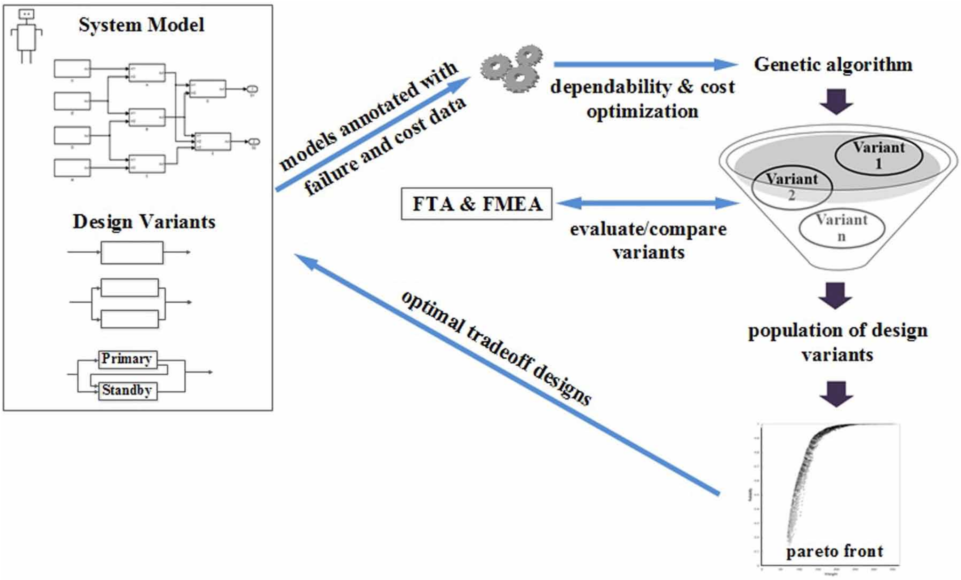
The reduction problem in this chapter consists of rewriting the sum of products that is generated from the failure behavioral model of the system into an equivalent sum of the relevant and minimized products. It is hoped that these results alleviate some of the problems outlined in (Rauzy, A., Châtelet, E., Dutuit, Y., & Bérenguer, C., 2003), and thereby making it easier to compute the probability of the original sum of products.

A related work on manipulation of algebraic expressions consists of a minimization algorithm which can be found in (Merle et al., 2010). The latter removes from the sum the product terms that are entirely redundant, which is necessary but not sufficient. Therefore, we present here a full minimization algorithm that removes the completely redundant product terms, minimizes the remaining terms, and removes the terms that become redundant in the process. The underlying idea and its usefulness in model reduction and design for robust system safety was outlined in (Mahmud, 2017). Also, the reduction algorithm developed in this chapter together with the generation and synthesis algorithms in (Mahmud, 2015) set the stage for a high-level of automation and integration of such advanced fault tree analysis with model-based design to effectively improve development of robotics systems.

Moreover, an advanced FTA which yields more accurate analysis results through consideration of the dynamic aspects can have a positive impact on a process aiming at cost-effectively improving dependability of robotic systems (see to the center of Figure 4). More precisely, reliability, which is one of the crucial aspects of dependability, can be increased iteratively via a process of optimization of design models. That is, given a robotic system model that is annotated with failure and cost information together with sets of design variants to the components; several variants can be used to implement a function or a component element, but each variant has different dependability characteristics and cost. The solutions for component redundancy allocations having the best trade-offs in dependability, cost and possibly other objectives including e.g. response time, can be selected automatically for the architecture. This has been made possible with advances in heuristic approaches (i.e., genetic algorithms) to multi-objective optimization of designs (Konak, Coit, & Smith, 2006; Mian et al., 2015). The aim is to arrive at solutions that are identified as Pareto optimal—i.e., in such a solution, any objective cannot be improved any further without compromising another one (e.g., safety, cost…).

In such a context for instance, an algorithm to determine minimal failure scenarios from the algebraic expressions, which are generated from the dependability-

Figure 4. Automatic optimization of design models



related annotations (i.e., SMs), facilitates the probabilistic assessments needed to compare variants. Failure probabilities at system level can also be computed through synthesis and analysis of the minimal failure sequences obtained from the component annotations.

The following Algorithm 1 takes a standard sum-of-product canonical form (*S*) as input then progressively reduces it into a minimized form (S_{min}). It starts by eliminating the product terms that are completely redundant (lines 3 and 4). If a PT is not redundant as a whole, the algorithm tries to reduce it by checking the overall relevance of every order-sensitive part (lines 6-15). This may introduce new redundancies that can be removed by using some Boolean idempotent and POR transformation laws (line 16 then use of Algorithm 2). The reduced PTs are saved in S_{min} (line 17). Moreover, further reductions can be made by removing the redundant sequences of a product term in S_{min} (lines 18-26). Consequently, a product which was not entirely redundant in the previous steps may become redundant due to possible changes to it or to other product terms—e.g., reducing a PT into its relevant cut-sequences. Therefore, line 30 (then use of Algorithm 3) removes such later-appearing redundancies from S_{min}.

Algorithm 1: S_{min} MinimalCanonicalForm(*S*)
Input: a standard sum-of-product canonical form
Output: a minimal standard sum-of-product canonical form

```
1:  let $S_{min} = S$
2:  for each PT in S do
   /* a PT is a cut-sequence set which is also (algebraically)
 a cut-sequence sum (CSS)*/
3:    if $PT \bullet \sum_{CSS_{\neq PT} \in S_{min}} CSS = PT$ then
4:       let $S_{min} = S_{min} \setminus PT$
5:  else
6:  let $PT_r = PT$
7:  let $PT_p = PT$
8:  for each $(E_j \mid E_k)$ in PT do
9:       replace $(E_j \mid E_k)$ with $E_j$ in $PT_r$
10:      if $PT_r \bullet \sum_{CSS \in S_{min}} CSS = PT_r$ then
        /* $PT_p$ is reduced to $PT_r$ */
```

```
11:     let $\mathrm{PT_p} = \mathrm{PT_r}$
12:     else
       /* $\mathrm{PT_r}$ is restored */
13:     let $\mathrm{PT_r} = \mathrm{PT_p}$
14:     end if
15: end for
16:     let $\mathrm{PT_r}$= CleanupByIdempTransfLaws($\mathrm{PT_r}$)
       /* PT is reduced in $S_{\min}$ */
17:     replace PT with $\mathrm{PT_r}$ in $S_{\min}$
18:     let $\mathrm{PT_r} = \bigcup\{\mathrm{CS}\}$
       /*$\mathrm{PT_r}$ is the set (algebraically sum) of its cut-sequences
(CSs) */
19:     let $\mathrm{PT_p} = \mathrm{PT_r}$
20:     for each CS in $\mathrm{PT_r}$ do
21:             if $\mathrm{CS} \bullet \sum_{\mathrm{CSS}_{\neq \mathrm{PT_r}} \in S_{\min}} \mathrm{CSS} = \mathrm{CS}$ then
22:                         let $\mathrm{PT_p} = \mathrm{PT_p} \setminus \{\mathrm{CS}\}$
23:                 end if
24:             end for
25:             if $\mathrm{PT_p} \neq \mathrm{PT_r}$ then
            /* reduce $\mathrm{PT_r}$ further in $S_{\min}$ */
26:             replace $\mathrm{PT_r}$ with $\mathrm{PT_p}$ in $S_{\min}$
27:   end if
28: end if
29: end for
30: let $S_{\min}$ = FinalProductTermRedundancyRemoval($S_{\min}$)
31: return $S_{\min}$
```

Algorithm 2 below takes a product term as parameter, then reduces it by using the Boolean idempotent law $\mathrm{E} \bullet \mathrm{E} \Leftrightarrow \mathrm{E}$ (lines 2-4) and the POR transformation laws $(\mathrm{E_j}|\mathrm{E_k}) \bullet \mathrm{E_j} \Leftrightarrow \mathrm{E_j} \bullet (\mathrm{E_j}|\mathrm{E_k}) \Leftrightarrow (\mathrm{E_j}|\mathrm{E_k})$ (lines 5-10).

Algorithm 2: PT CleanupByIdempTransfLaws(PT)
Input: a product term PT
Output: the PT is reduced by absorption

```
let lPT = PT
for each $E_j \bullet E_j$ in lPT do
    replace $E_j \bullet E_j$ with $E_j$ in lPT
end for
for each $(E_j|E_k) \bullet E_j$ in lPT do
    replace $(E_j|E_k) \bullet E_j$ with $(E_j|E_k)$ in lPT
end for
for each $E_j \bullet (E_j|E_k)$ in lPT do
    replace $E_j \bullet (E_j|E_k)$ with $(E_j|E_k)$ in lPT
end for
return lPT
```

The following Algorithm 3 is used to cleanup $S_{\min}$ from the product terms that become redundant during the reduction process. It takes a standard sum of product canonical form as parameter, then removes the entirely redundant product terms from the sum.

Algorithm 3: *S* FinalProductTermRedundancyRemoval(*S*)
Input: a standard sum of product canonical form $S = \sum \mathrm{PT}$
In term of sets,
$S = \bigcup \mathrm{PT}$, each product term PT is a cut sequence set (CSS)
Output: the canonical form without the redundant product terms

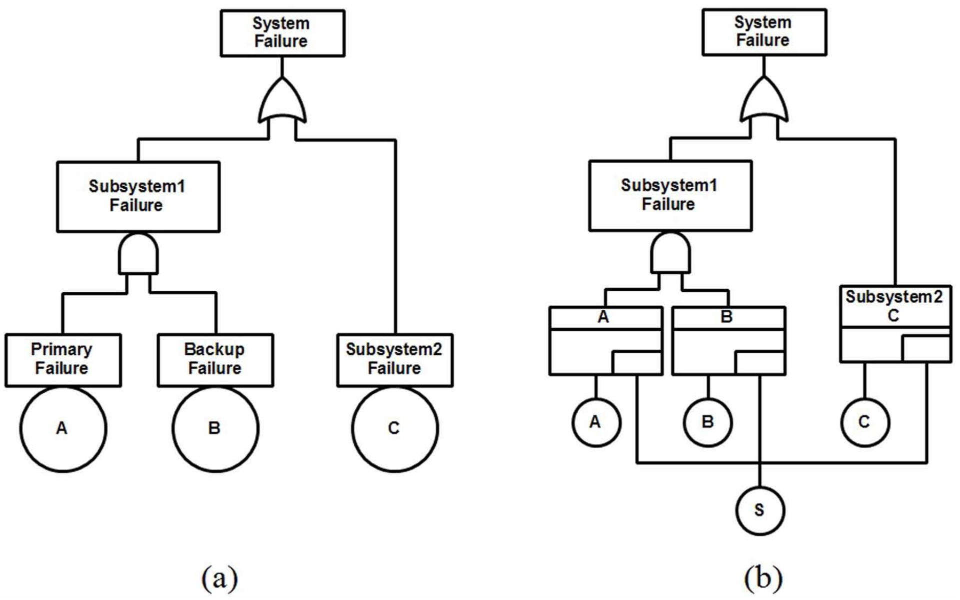
```
let lS = S
for each PT in lS do
    if $\mathrm{PT} \bullet \sum_{\mathrm{CSS}_{\neq \mathrm{PT}} \in \ell S} \mathrm{CSS} = \mathrm{PT}$ then
        let $\ell S = \ell S \setminus \mathrm{PT}$
    end if
end for
return lS
```

CASE STUDY

For instance, let us consider the example adapted from (Visinsky et al., 1993) of two redundant sensors in a robot joint. These are the primary and backup sensors which compose Subsystem1 of Figure 5a (i.e., the fault tree representation of the system). The basic events A and B (denoted by circles) correspond to the failures of primary and backup sensors, respectively. Subsystem1 fails if both sensors fail, hence the logical AND gate on the left hand side of the figure. Moreover, the motion at the joint is coupled with another motion (Subsystem2) such that failure of either motion causes a joint total failure, and hence the logical OR gate to the top of the figure. However, Subsystem2 is not supplied with a fault tolerance mechanism. Thus, its basic event C is a single point of failure.

To increase the reliability of the system, the actively redundant sensors of Subsystem1 and the sensor of Subsystem2 all share the same spare unit S, as in Figure 5b—assuming that they are all functionally equivalent. In such configuration, C is no longer a single point of failure. We wish to note, though, that this is not an uncommon pattern in safety-critical systems as it can represent a redundancy which is not necessarily of a kinematic type—i.e., a redundancy of components (i.e., actuators and sensors) or a redundancy in the control architecture which can allow better performance than ordinary kinematic controllers (Avanzini, Ceriani, Zanchettin, Rocco, & Bascetta, 2014; Flacco & De Luca, 2015). Moreover, an example of use of such pattern can be found in aerospace applications, like the vehicle management example system described in (Vesely, Stamatelatos, Dugan, Fragola, Minarick, & Railsback, 2002).

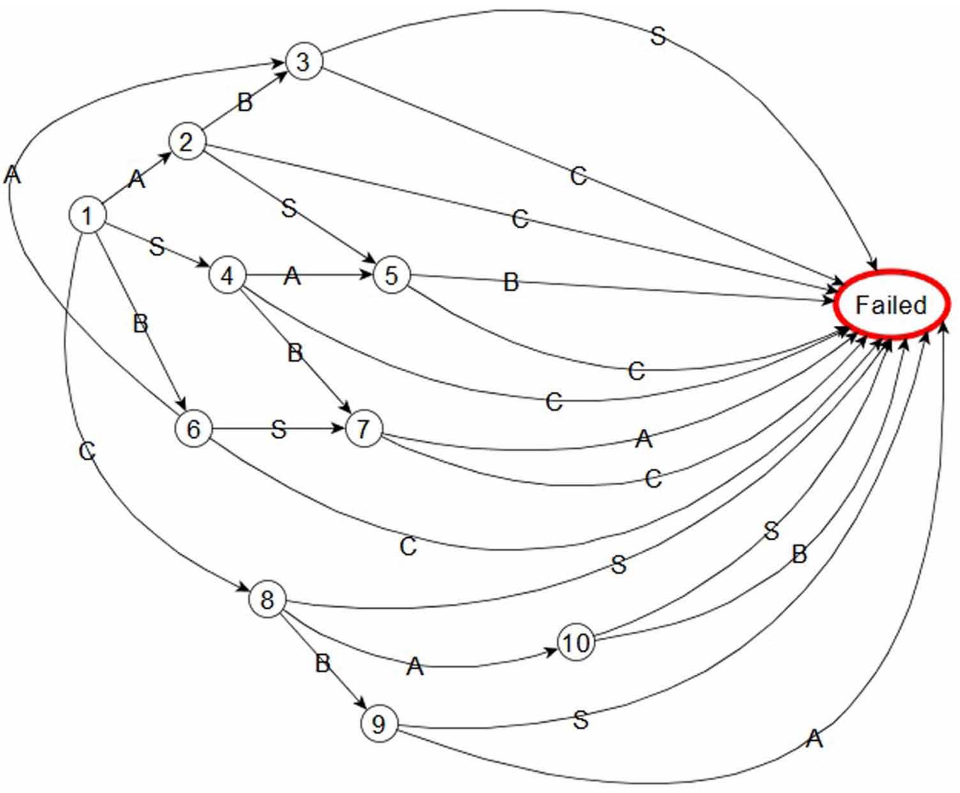
Figure 5. (A) Fault tree of a robot joint sensing system (RJSS) with active redundancy and (b) with an extra passive redundancy



For the purpose of this study, this example system (Figure 5b) is useful to demonstrate the reduction potential of the minimization algorithm proposed in the previous section (which is a central contribution of this chapter). Unlike the previous primary-standby example system with its 6 PTs, the failure behavioral model of the RJSS (Figure 6) gives 20 product terms, out of which 5 are initially entirely redundant and 3 will be removed after reduction into redundant cut-sequences. The remaining 12 PTs are minimized into their corresponding relevant cut-sequences, thereby giving a minimal canonical form which can be represented ultimately by a sum of 5 product terms.

In the example system, S is supposed to take over any unit (A, B, or C) whichever fails first. In case a unit fails but cannot be rescued by the spare, the output event of the corresponding spare gate to the middle of the figure becomes true (i.e., a fault event). A spare unit cannot take over a failed unit if it is also failed or is not available (e.g., it took over another unit that failed earlier). Such capability can be useful to implement cost-effective fault tolerance solutions that meet dependability requirements, especially if the sensors are unlikely to fail.

Figure 6. Failure behavioral model of the RJSS



Now the order in which failure events occur can affect the overall outcome (i.e., total failure of the system). For example, if A fails first, it will be rescued by the spare, then if C fails second, in this case the spare is no longer available to take over C, and thereby causing a total failure. However, if C fails first, then it will be rescued by S, but a subsequent failure of A is not severe since A is in active redundancy with B.

Figure 6 represents the failure behavior of the RJSS. Transitions are labelled by internal failures of components A, B, C, and shared spare S. Initial state 1 is a state in which none is failed. Also, states 2 through 10 are states in which the system is degraded but still operational. However, 'Failed' is a state in which there is total loss of function.

The algorithm that generates failure expressions from state machines, which is outlined in (Mahmud, 2015) and summarized in the background section, produces the following product terms numbered 1 through 20 in Table 2. For the sake of clarity, a special focus was put on some iterations; namely, iteration #0 (initial status), a

Table 2. Excerpt of reduction iterations

<table>
<tr><th>PT#</th><th>#0</th><th>MS#1</th><th>MS#2</th><th>MS#3</th><th>minimal</th></tr>
<tr><td>1</td><td><b>S.A|B|S.C|S|B|A</b> +</td><td><b>C.A.S</b> +</td><td><b>C&lt;A&lt;S</b> +</td><td><b>C&lt;A&lt;S</b> +</td><td><b><i>removed</i></b></td></tr>
<tr><td>2</td><td>C|S|A.B|C|S|A +</td><td>C.B|C +</td><td>B&lt;C +</td><td>B&lt;C +</td><td>B&lt;C +</td></tr>
<tr><td>3</td><td>S|B|A.C|S|B|A +</td><td>C.S +</td><td>C&lt;S +</td><td>C&lt;S +</td><td>C&lt;S +</td></tr>
<tr><td>4</td><td><b>C.B|S|C.A|C|S|B</b> +</td><td><b>C.B|S|C.A|C|S|B</b> +</td><td><b>C.B|S|C.A|C|S|B</b> +</td><td><b><i>removed</i></b></td><td><b><i>removed</i></b></td></tr>
<tr><td>5</td><td><b>C.A|S|C.B|C|S|A</b> +</td><td><b>C.A|S|C.B|C|S|A</b> +</td><td><b>C.A|S|C.B|C|S|A</b> +</td><td><b><i>removed</i></b></td><td><b><i>removed</i></b></td></tr>
<tr><td>6</td><td>S.B|S.A|C|S|B +</td><td>S.B|S.A|C|S|B +</td><td>S.B|S.A|C|S|B +</td><td>A&lt;B&lt;S +</td><td>A&lt;B&lt;S +</td></tr>
<tr><td>7</td><td>S.A|S.B|C|S|A +</td><td>S.A|S.B|C|S|A +</td><td>S.A|S.B|C|S|A +</td><td>B&lt;A&lt;S +</td><td>B&lt;A&lt;S +</td></tr>
<tr><td>8</td><td>B.A|B.S|C|B|A +</td><td>B.A|B.S|C|B|A +</td><td>B.A|B.S|C|B|A +</td><td>S&lt;A&lt;B +</td><td>S&lt;A&lt;B +</td></tr>
<tr><td>9</td><td>B.S|B.A|C|S|B +</td><td>B.S|B.A|C|S|B +</td><td>B.S|B.A|C|S|B +</td><td>A&lt;S&lt;B +</td><td>A&lt;S&lt;B +</td></tr>
<tr><td>10</td><td>A.B|A.S|C|B|A +</td><td>A.B|A.S|C|B|A +</td><td>A.B|A.S|C|B|A +</td><td>S&lt;B&lt;A +</td><td>S&lt;B&lt;A +</td></tr>
<tr><td>11</td><td>A.S|A.B|C|S|A +</td><td>A.S|A.B|C|S|A +</td><td>A.S|A.B|C|S|A +</td><td>B&lt;S&lt;A +</td><td>B&lt;S&lt;A +</td></tr>
<tr><td>12</td><td><b>C.A|B|C.S|C|B|A</b> +</td><td><b>C.A|B|C.S|C|B|A</b> +</td><td><b>C.A|B|C.S|C|B|A</b> +</td><td><b>S&lt;A&lt;C</b> +</td><td><b><i>removed</i></b></td></tr>
<tr><td>13</td><td><b>C.S|C|B.A|C|S|B</b> +</td><td><b>C.S|C|B.A|C|S|B</b> +</td><td><b>C.S|C|B.A|C|S|B</b> +</td><td><b>A&lt;S&lt;C</b> +</td><td><b><i>removed</i></b></td></tr>
<tr><td>14</td><td>C|B|A.S|C|B|A +</td><td>C|B|A.S|C|B|A +</td><td>C|B|A.S|C|B|A +</td><td>S&lt;C +</td><td>S&lt;C +</td></tr>
<tr><td>15</td><td>C|S|B.A|C|S|B +</td><td>C|S|B.A|C|S|B +</td><td>C|S|B.A|C|S|B +</td><td>A&lt;C +</td><td>A&lt;C +</td></tr>
<tr><td>16</td><td><b>C.B|A|C.S|C|B|A</b> +</td><td><b>C.B|A|C.S|C|B|A</b> +</td><td><b>C.B|A|C.S|C|B|A</b> +</td><td><b><i>removed</i></b></td><td><b><i>removed</i></b></td></tr>
<tr><td>17</td><td><b>C.S|C|A.B|C|S|A</b> +</td><td><b>C.S|C|A.B|C|S|A</b> +</td><td><b>C.S|C|A.B|C|S|A</b> +</td><td><b><i>removed</i></b></td><td><b><i>removed</i></b></td></tr>
<tr><td>18</td><td><b>S.B|A|S.C|S|B|A</b> +</td><td><b>S.B|A|S.C|S|B|A</b> +</td><td><b>S.B|A|S.C|S|B|A</b> +</td><td><b><i>removed</i></b></td><td><b><i>removed</i></b></td></tr>
<tr><td>19</td><td>A.B|A.C|S|B|A +</td><td>A.B|A.C|S|B|A +</td><td>A.B|A.C|S|B|A +</td><td>C&lt;B&lt;A +</td><td>C&lt;B&lt;A +</td></tr>
<tr><td>20</td><td>B.A|B.C|S|B|A</td><td>B.A|B.C|S|B|A</td><td>B.A|B.C|S|B|A</td><td>C&lt;A&lt;B</td><td>C&lt;A&lt;B</td></tr>
</table>

step (milestone MS#1), two other steps (MS#2 and MS#3), and finally the results (minimal sequences).

The column #0 shows all PTs which were produced by the generation algorithm. In MS#1, the order-sensitive parts of, e.g., PT#1 (i.e. A|B|S.C|S|B|A <=> (A|B).(A|S).(C|S).(C|B).(C|A)) were ultimately reduced into the corresponding order-insensitive parts (e.g., from (A|B). (A|S).(C|S).(C|B).C to (A|B). (A|S).(C|S).(C|B) to…. to ultimately A.C) by Algorithm 1 (lines 8-15) due to their overall irrelevance (condition in line 10 satisfied) and by Algorithm 2 by using the Boolean idempotent and POR transformation laws. In MS#2, for the same case of PT#1 for instance, the obtained logical conjunction C.A.S is a set of 6 sequences out of which only C<A<S (i.e., the sequence CAS) is preserved by Algorithm 1 (lines 20-24). This sequence will be finally removed by Algorithm 3 as it has become redundant in line 30 of Algorithm 1. In this sense, this case is similar to the cases of PT#12 and PT#13 in which S<A<C and A<S<C have likewise been removed by Algorithm 3. Unlike these examples, PT#4, PT#5, PT#16, PT#17 and PT#18 were removed much earlier and without further processing as they were initially entirely redundant.

The minimal sequences produced by Algorithm 1 appear in the rightmost column. A minimal sequence being a sequence such that if we retrieve any event it would no longer be accepted by the state machine. Moreover, if we have n events out of which all n! sequences appear each as a minimal result, then the temporal order is insignificant and, therefore, the conjunction of these events can be presented as one minimal result. For example, the product terms numbered 6 through 11 can be represented as a single minimal result (i.e., the conjunction A.B.S) —as all sequences made out of A, B and S are present. Similarly, PT#3 and PT#14 can be merged into the conjunction S.C.

This has demonstrated that Algorithm 1 ultimately yields the following result for the global failure condition, and without any loss of the significance of the event-order:

Total failure = S.C + A<C + B<C + S.A.B + (C<A).(C<B)

This result shows that the system fails due to any of the following causes: both S and C fail; A fails then C fails; B fails then C fails; A, B and S all fail; or A and B both fail after failure of C.

Despite its few components, the example system used exhibits very complex dependencies. For large systems, this algorithm is practical via composition. That is, failure expressions local to the system constituents are generated from their corresponding state machines, reduced as outlined in the algorithm, and then synthesized into expressions of system failure conditions which likewise can be reduced further if necessary (Mahmud, 2012, 2015). The algorithm that we developed in this chapter is a significant contribution to model reduction (Mahmud, 2017).

FUTURE RESEARCH DIRECTIONS

Automation of the generation, synthesis, and analysis of fault trees that are extended with sequencing of events from robotic models (e.g. RobotML) as part of integrated model-based design and analysis processes is an open issue. Moreover, there is still a lack of tools for the automated analysis of this type of fault trees. This is mainly due to the complexity of the manipulation as well as probabilistic assessment of the corresponding expressions. This chapter presents, therefore, a contribution towards a higher degree of automation in this direction.

On another note, repairable events would be worth being considered as many users aim at modeling repairable systems. We wish to note, though, that the approach in this chapter does not totally preclude the existence of cyclic behaviors (failures and repairs)—i.e., after the transformation of the embedded acyclic behaviors into algebraic expressions, and the reduction thereof, the obtained results represent, therefore, minimal circuitless sequences onto which are grafted the repair circuits. However, this would imply analysis of (coherent) systems with perfect repair policy. Repairable systems with different maintenance policies are likewise worth being considered in future extensions of this work.

CONCLUSION

In this chapter, we outlined a lack of suitable techniques that generate, synthesize and analyze fault trees with the significant sequencing of events from models specific to robotics systems. Most of the existing work deals with static fault trees; however, less efforts are on generation of dynamic fault trees but with a main focus on quantitative analysis. In this regard, this chapter puts emphasis on such important issue by highlighting effects of sequence-dependent failures and the importance of an adequate qualitative analysis that reduces some of the difficulties encountered in probabilistic assessments. The second contribution is a solution proposed for the reduction of fault trees with more expressive power and that are generated from automata descriptions of failure behaviors. In several safety-critical domains, failure behavioral models are increasingly automata based like with EAST-ADL in the automotive industry, Altarica and AADL in aerospace, and RobotML in the robotic domain. Therefore, the algorithms proposed in this chapter complement the previous contributions regarding advanced generation and synthesis of such fault tree expressions. These allow improved failure prediction capabilities which should be used as an input to the design of proactive failure avoidance measures. Likewise, as redundancy allocation is a popular reliability improvement technique, we also presented the usefulness of this work as part of an approach to cost-effectively improving dependability of the overall robotic system via a process of multi-objective optimization of design models.

REFERENCES

Avanzini, G. B., Ceriani, N. M., Zanchettin, A. M., Rocco, P., & Bascetta, L. (2014). Safety control of industrial robots based on a distributed distance sensor. *IEEE Transactions on Control Systems Technology*, *22*(6), 2127–2140. doi:10.1109/TCST.2014.2300696

Boudali, H., Crouzen, P., & Stoelinga, M. (2010). A rigorous, compositional, and extensible framework for dynamic fault tree analysis. *Dependable and Secure Computing*. *IEEE Transactions on*, *7*(2), 128–143.

Chen, D., Mahmud, N., Walker, M., Feng, L., Lönn, H., & Papadopoulos, Y. (2013). Systems Modeling with EAST-ADL for Fault Tree Analysis through HiP-HOPS. In *Proceedings of the 4th International Federation of Automatic Control (IFAC) Workshop on Dependable Control of Discrete Systems (DCDS), York*, (pp. 91-96). IFAC. 10.3182/20130904-3-UK-4041.00043

Crouzen, P., Hermanns, H., & Zhang, L. (2008). On the minimisation of acyclic models. In *International Conference on Concurrency Theory* (pp. 295-309). Springer Berlin Heidelberg.

Dehlinger, J., & Dugan, J. B. (2008). Analyzing dynamic fault trees derived from model-based system architectures. *Nuclear Engineering and Technology: An International Journal of the Korean Nuclear Society*, *40*(5), 365–374. doi:10.5516/NET.2008.40.5.365

Dugan, J. B., Bavuso, S. J., & Boyd, M. A. (1992). Dynamic fault-tree models for fault-tolerant computer systems. *Reliability*. *IEEE Transactions on*, *41*(3), 363–377.

EAST-ADL. (2010). *EAST-ADL Domain Model Specification, D4.1.1*. Retrieved from http://www.atesst.org/home/liblocal/docs/ATESST2_D4.1.1_EAST-ADL2-Specification_2010-06-02.pdf

Flacco, F., & De Luca, A. (2015). Discrete-time redundancy resolution at the velocity level with acceleration/torque optimization properties. *Robotics and Autonomous Systems*, *70*, 191–201. doi:10.1016/j.robot.2015.02.008

IEEE Std.352. (1987). *IEEE Guide for General Principles of Reliability Analysis of Nuclear Power Generating Station Safety Systems.* ANSI/IEEE.

Joshi, A., Vestal, S., & Binns, P. (2007). *Automatic generation of static fault trees from aadl models*. In Workshop on Architecting Dependable Systems of The 37th Annual IEEE/IFIP Int. Conference on Dependable Systems and Networks, Edinburgh, UK.

Karbasian, M., Mehr, M. R., & Agharajabi, M. (2012). Improvement of the Reliability of Automatic Manufacture Systems by Using FTA Technique. *International Journal of Industrial Engineering*, *23*(2), 85–89.

Kolagari, R. T., Chen, D., Lanusse, A., Librino, R., Lönn, H., Mahmud, N., ... Yakymets, N. (2015). Model-Based Analysis and Engineering of Automotive Architectures with EAST-ADL: Revisited. *International Journal of Conceptual Structures and Smart Applications*, *3*(2), 25–70. doi:10.4018/IJCSSA.2015070103

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms. *Reliability Engineering & System Safety*, *91*(9), 992–1007. doi:10.1016/j.ress.2005.11.018

Mahmud, N. (2012). *Dynamic Model-based Safety Analysis: From State Machines to Temporal Fault Trees* (Ph.D. dissertation). Department of Computer Science, University of Hull, Hull, UK.

Mahmud, N. (2015). Advanced fault tree synthesis for systems with dynamic aspects. In *Safety and Reliability of Complex Engineered Systems* (pp. 1635–1643). CRC Press. doi:10.1201/b19094-213

Mahmud, N. (2015). Improving Dependability of Robotics Systems, Experience from Application of Fault Tree Synthesis to Analysis of Transport Systems. *International Journal of Robotics Applications and Technologies*, *3*(2), 38–62. doi:10.4018/IJRAT.2015070103

Mahmud, N. (2017). A compositional symbolic calculus approach to producing reduced Markov chains. In *2017 Annual Reliability and Maintainability Symposium (RAMS)*. IEEE. 10.1109/RAM.2017.7889660

Mahmud, N., & Mian, Z. (2013). Automatic generation of temporal fault trees from AADL models. In Safety, Reliability and Risk Analysis: Beyond the Horizon (pp. 2741–2749). CRC Press. doi:10.1201/b15938-415

Mahmud, N., Papadopoulos, Y., & Walker, M. (2010). A Translation of State Machines to Temporal Fault Trees. In *Proceedings of the 40th IEEE/IFIP International Conference on Dependable Systems and Networks*, (pp. 45-51). IEEE. 10.1109/DSNW.2010.5542620

Mahmud, N., Walker, M., & Papadopoulos, Y. (2012). Compositional synthesis of temporal fault trees from state machines. *Performance Evaluation Review*, *39*(4), 79–88. doi:10.1145/2185395.2185444

Merle, G., Roussel, J. M., Lesage, J., & Bobbio, A. (2010). Probabilistic algebraic analysis of fault trees with priority dynamic gates and repeated events. *Reliability. IEEE Transactions on*, *59*(1), 250–261.

Mian, Z., Bottaci, L., Papadopoulos, Y., Sharvia, S., & Mahmud, N. (2015). Model Transformation for Multi-objective Architecture Optimisation of Dependable Systems. In *Dependability Problems of Complex Information Systems* (pp. 91–110). Springer. doi:10.1007/978-3-319-08964-5_6

Murray, L., Liu, W., Winfield, A., Timmis, J., & Tyrrell, A. (2012). Analysing the Reliability of a Self-reconfigurable Modular Robotic System. In Bio-Inspired Models of Networks, Information, and Computing Systems (pp. 44-58). Springer Berlin Heidelberg. doi:10.1007/978-3-642-32711-7_4

Rauzy, A. (2002). Mode automata and their compilation into fault trees. *Reliability Engineering & System Safety*, *78*(1), 1–12. doi:10.1016/S0951-8320(02)00042-X

Rauzy, A., Châtelet, E., Dutuit, Y., & Bérenguer, C. (2003). A practical comparison of methods to assess sum-of-products. *Reliability Engineering & System Safety*, *79*(1), 33–42. doi:10.1016/S0951-8320(02)00165-5

Vesely, W., Stamatelatos, M., Dugan, J., Fragola, J., Minarick, J. III, & Railsback, J. (2002). *Fault Tree Handbook with Aerospace Applications*. NASA Office of Safety and Mission Assurance.

Vesely, W. E. (1981). *Fault Tree Handbook.* US Nuclear Regulatory Committee Report, NUREG-0492.

Visinsky, M. L., Walker, I. D., & Cavallaro, J. R. (1993). Robotic fault tolerance: algorithms and architectures. *Robotics and Remote Systems in Hazardous Environments*, 53-73.

Walker, M., & Papadopoulos, Y. (2009). Qualitative temporal analysis: Towards a full implementation of the Fault Tree Handbook. *Control Engineering Practice*, *17*(10), 1115–1125. doi:10.1016/j.conengprac.2008.10.003

Yakymets, N., Dhouib, S., Jaber, H., & Lanusse, A. (2013). Model-driven safety assessment of robotic systems. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on* (pp. 1137-1142). IEEE.

KEY TERMS AND DEFINITIONS

AADL: AADL is an architecture analysis and design language standardized in 2004 by the Society of Automotive Engineers. It is used in the specification and analysis of the software and hardware architecture of real time embedded systems. AADL is mainly devoted to performance-critical aerospace and automotive systems.

Dynamic Fault Tree: Dynamic fault tree (DFT) was invented in response to a shortage in modelling sequence-dependent failures by using standard combinatorial fault trees. It thus added capabilities for capturing the dynamic aspects exhibited by modern complex systems. The DFT is designed primarily for quantitative analysis, in general by using Markov techniques. However, there has been less focus on qualitative analysis at the level of the DFT, i.e. determination of its (minimal) cut-sequences.

EAST-ADL: EAST-ADL is an electronic architecture and software technology architecture description language for automotive embedded systems, which was developed by a consortium of universities and automotive companies. The language was further refined within the framework of the model-based analysis and engineering of novel architectures for dependable electric vehicles (MAENAD) EU FP7 project. Aspects covered by EAST-ADL include vehicle features, functions, requirements, variability, dependability, software components, hardware components and communication.

Failure Mode and Effects Analysis: Failure mode and effects analysis (FMEA) is commonly known (and often used) as a bottom-up analysis technique. It proceeds by analyzing the system components individually, or sometimes collectively, to inductively derive the consequences of their failures on the system. FMEA aims at addressing those effects and the technique is widely practiced in reliability engineering in high-hazard industries.

Fault Tree Analysis: Fault tree analysis (FTA) postulates a hazard (top event of the fault tree) which must be avoided. It then reasons backward to identify all logical combinations of events which could lead the system to that hazard. FTA can be quantitative by combining figures for component failure rates to calculate overall probabilities of system hazards. It can also be qualitative by eliminating the component failures that are irrelevant to the total failure of the system, and thereby attempting to produce minimal failure scenarios. FTA is widely practiced in reliability engineering and in high-hazard industries.

Fault-Tolerance Systems: Fault-tolerance enables a system to continue its intended operation in the event of faults in some parts of it. Fault-tolerance systems rely typically on redundant units. The redundant components can be active and operating in parallel, or passive but switched into active use upon failure of the primary units. They can also be designed to be shared by a number of functionally identical systems.

State Automata to Fault Trees: State automata to fault trees with order-dependent behaviors (SAFORA) is a top-down deductive synthesis approach that computes expressions of global failure conditions from the dysfunctional behavior local to the components of the system (described as hierarchical state machines). The purpose is to accurately analyze the reliability of architectures and the technique is suitable for complex systems featuring dynamic aspects.

Chapter 6
History of Service Robots and New Trends

Teresa T. Zielinska
Warsaw University of Technology, Poland

ABSTRACT

The chapter gives an overview of the current developmental trends in service robotics. First the short history of service robots with its precursors is presented together with the definitions. The developmental trends and statistical data are summarized. The overview of service robots includes the ancient robot precursors, the middle ages, and the period of industrial revolution. The representative examples of different kinds of service robots built in the 20th and 21st centuries are also given. The chapter is concluded by discussing the perspectives of service robotics.

INTRODUCTION

Till the year 2012 the term service robots had no strict officially accepted definition: they had very different structures, capabilities and applications.

By the end of the nineties the International Service Robot Association (ISRA) issued the following working definition of service robots: "Machines that sense, think, and act to benefit or extend human capabilities and to increase human productivity" (Pransky, 1996).

International Federation of Robotics (IFR) gave the following provisional definition: "A service robot is a robot which operates semi- or fully autonomously to perform services useful to the well-being of humans and equipment, excluding manufacturing operations".

DOI: 10.4018/978-1-5225-5276-5.ch006

And it continued with explanation:

With this definition, manipulating industrial robots could also be regarded as service robots, provided they are installed in non-manufacturing operations. Service robots may or may not be equipped with an arm structure as is the industrial robot. Often, but not always, the service robots are mobile. In some cases, service robots consist of a mobile platform on which one or several arms are attached and controlled in the same mode as the arms of the industrial robot.

In 1999 International Federation of Robotics (IFR) and the United Nations Economic Commission for Europe for the first time included service robots in statistical reports. The fast development of this class of robots caused, that in 2008 IFR decided to split their World Robotics Yearbook containing robot statistics into two volumes, with one dedicated only to the service robots.

The effort directed at the terminology unification was started in 1995 by the United Nations Economic Commission for Europe (UNECE) and IFR: it resulted in novel ISO-Standard 8373 definition which became effective in the year 2012, it states:

“A robot is an actuated mechanism programmable in two or more axes with a degree of autonomy, moving within its environment, to perform intended tasks. Autonomy in this context means the ability to perform intended tasks based on current state and sensing, without human intervention.

A service robot is a robot that performs useful tasks for humans or equipment, excluding industrial automation application. This statement matches the ISO 8373 definition of service robot by The International Organization for Standardization. A personal service robot or a service robot for personal use is a service robot used for a non-commercial task, usually by lay persons. Examples are: domestic servant robot, automated wheelchair, personal mobility assistive robot, or pet exercising robot. A professional service robot or a service robot for professional use is a service robot used for a commercial task, usually operated by a properly trained operator. Examples are: cleaning robot used in public places, delivery robot in offices or hospitals, fire-fighting robot, rehabilitation robot or surgery robot in hospitals. In this context an operator is a person designated to start, monitor and stop the intended operation of a robot or a robot system.

The above classification takes into account the type of the user (professional versus personal user) as a classification criterion. Another possible robot categorisation takes into account the type of the environment the robot acts in (Zielinski, 2010):

Industrial robots operate in a fully structured environment. For example, in the work cell all devices are precisely located, so it is sufficient that robot control is position based, and therefore not many external sensors are needed.

Personal service robots operate in a quasi-structured environment, created by man for his own purposes. This means that the surroundings are not exactly adjusted to the needs of the performed job (e.g., regular home, waiting hall, office, restaurant).

Field robots work for anonymous recipients in the natural environment, which is fully unstructured - for example: forest, airspace, sea bottom, ruins, mountains. Field robots usually are associated with professional service robots.

In general, the actions of service robots are determined by the information gathered by external sensors.

With the fast development of various robots and widening of their application areas, observation made by Joseph Engelberger, the "father" of robotics, is still valid: "I can't define a robot, but I know one when I see one". Joseph Engelberger predicted that service robots one day will become the largest group, outnumbering the industrial robots by several times. Now this is becoming the true. The idea of helping humans in tedious or repetitive work, by artificial means has existed since the beginning of humanity. Thus, tools and machines were conceived, built and used as intermediate solutions with increasing performances over time.

Taking into account the type of performed task the following four categories of service robots can be distinguished (Zielinski, 2010):

- Professional service providers (acting on the ground, in the offices, in hospitals),
- Domestic service robots (helping in personal work, operating in the house, robots for entertainment and education),
- Security robots (working for defence, safety and rescue),
- Space robots (working in space or doing planet exploration).

The current developmental trends indicate that soon the service robots ("serving us" robots) will have millions of end-users in houses, hospitals, restaurants, offices, airports, etc.

SERVICE ROBOT MARKET

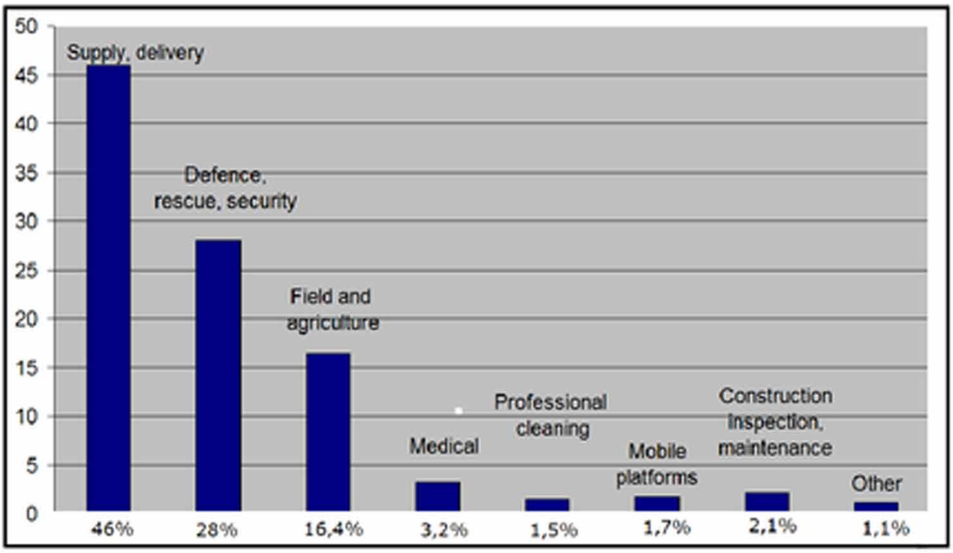
During the past century life expectancy rose significantly throughout the world. For wealthiest populations from around 50 to over 75 years. There were many reasons of such change – the quality of medical service has improved, the improvement of the working and living conditions as well as the awareness of the population of the healthy life style. In many developed countries the birth rate dropped, what caused the statistical increase of the average age of population and resulted in increased ratio of the elderly to the young people. As the result the human society faced new problems – the need for serving people using a limited work force. Additionally, with the increase in life comfort the employees are less willing to take on the boring and repetitive work, including the house chores and caretaking. Frequent occurrence of military conflicts and acts of terror or natural disasters also increased the demand for special robots, helping in dangerous missions and protecting the human life. All those factors motivate, while the technical and research progress make possible the development of service robots.

In the year 1989 Joseph Engelberger published the book “Robotics in Service” (Engelberger, 1989) where he forecasted a coming of service robot boom, which now takes place. this is justified by the market demand. According to the IFR Statistical Department in 2012 the overall number of professional service robots approached 126 000 units (IFR Statistical Department, 2013), and in the year 2014 it was already 172 000 (IFR Statistical Department, 2014) with the increase of 36%. In 2011 and 2012 annual sales of professional service robots was reported to be 15 000 units, in 2013 it raised to 24 207 units increasing by 60%. In 2012 IFR reported that the biggest number (in units) of sold professional robots by category was the number of the defence robots – 47%, similar trend was in 2013 and 2014 with the robots for defence applications accounting for 45% of the total professional service robots sold in the year 2014. The remaining sold professional service robots were mainly the supply and delivery robots. In the year 2015 the reported sales were 41 060 units, what is about 24% more in comparison with almost 33 000 units in 2014. The highest number of sold robots by category was that of the robots for supply and delivery (46%) and next came the robots for defence, rescue and security (28%) - Figure 1. For both types of robots IFR notes that the real numbers of deployed robots can be even significantly higher. According to the available data in the year 2015 the most significant sales’ increase was observed for supply and delivery robots - it is expected that such trend will be sustained.

After a small decrease of the number of medical robots sold in 2014 (by 5%), in the year 2015 the increase by 7% compared to the year 2014 was recorded. Similarly, as in previous years, the most important applications were the robots for assisting surgery and the robots for therapy The IFR emphasizes that the medical robots are the most costly service robots with an average unit price of about one million USD including accessories and services.

In the coming years the demand for supply and delivery robots will be growing, the increased demand for cleaning robots and mobile platforms is also expected In the past the market for general purpose mobile platforms was low, but this product will be in higher demand. The group of users who are not interested in mobile base prototyping and prefer robust and tested devices is increasing. Those users prefer to

Figure 1. The annual sale of different types of professional service robots in the year 2015 (diagram created on the basis of IFR Executive Summary World Robotics 2016 Service Robots).

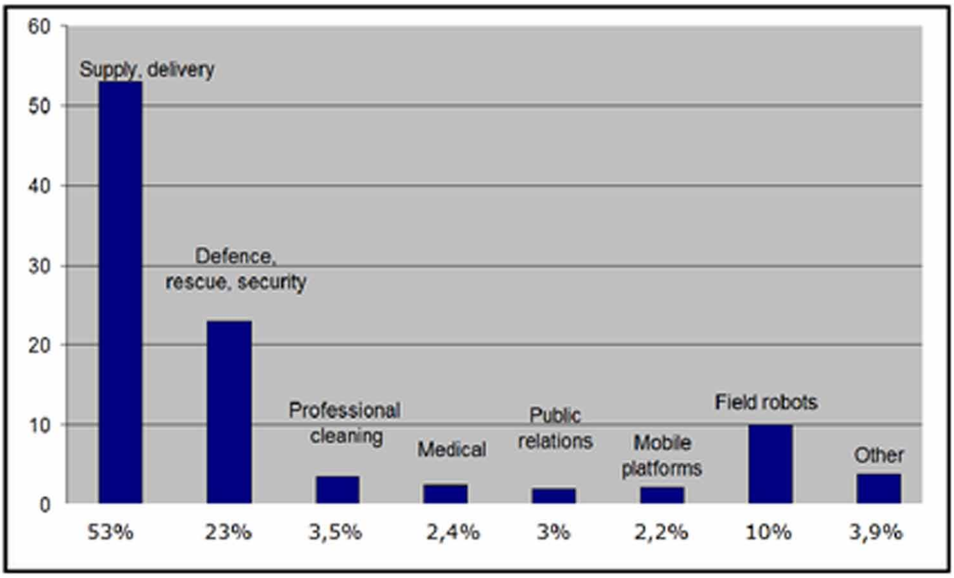


focus on sensing technologies and motion planning methods. The increased demand in the public relations/guiding robots (used in museums, exhibitions, supermarkets) is forecasted – Figure 2.

IFR surveys consider service robots for personal use separately. It is not possible to monitor the statistics of those robots accurately, as they are produced by many large and small vendors, from whom data collection is a huge problem. The unit value of personal robot is much smaller than that of a professional robot. The sale of personal robots in terms of units is very high, but their unit cost is low. The popular robotic toys or household robots are often very simple and their classification as robots is overly optimistic. It is obvious that in the future personal robots will be more capable. The household robots will specialise in caretaking and attending to the home chores. They will be able to perform dexterous manipulation and they will act autonomously in dynamic environments. This will be achieved with the support of advanced sensors and with the use of efficient perception methods. Such robots will profit from the advances of information technologies.

In the group of personal robots the sales of household and entertainment robots will further increase. From this point of view the future of robotics, attributed some time ago to low cost systems and innovative solutions to new applications (Ceccarelli, 2001), (Ceccarelli, 1998) is still valid. Such expectation is confirmed by statistical

Figure 2. Demand forecast for professional service robots in the nearest years – percentage considering the number of units (diagram based on the IFR data (IFR Statistical Department, 2016)).



data. In 2014, about 4.7 million of service robots for personal and domestic use were sold, what was 28% more than in the year 2013, and in the year 2015 about 5.4 million service robots for personal and domestic use was sold, what was 15% more than in 2014. Almost 65% of this amount were the domestic, mainly cleaning robots, another big quantity were the robotic toys, which contributed to more than 31% of the sold units. The robotic toys market is one of most expanding ones. In 2015 the sales grew by 29%, comparing to the year 2014. An increase in demand for robot companions and assistants (in that humanoids) is still expected, however such tendency has not been observed as yet. Similarly as in previous the years the marked will be dominated by the household, leisure and entertainment robots (Figure 3a). Referring to previous expectations, the market dynamics for leisure and entertainment robots will be a bit lower (Figure 3b), however for both household and entertainment robots it will be one of the most rapidly increasing (Figure 3c).

The majority of robot producers is located in Europe, but North America is also a significant deliverer. The Asian production is by 50% lower than Amercian (Figure 4), but we should not forget about the technological potential of this continent. For example, Japan leads in inventing novel types of service robots, and the fast developing Chinese industry needs not only the typical manipulators, but dedicated robotic systems. Booming Chinese economy creates an important market for service robots

Figure 3. Demand forecast for personal service robots for the nearest years - percentage considering the units count (a), forcasted in 2015 increase of unit sales related to the year 2014 (b), expected increase of sales related to the year 2015 (c) (diagrams based on the IFR data (IFR Statistical Department, 205, 2016).

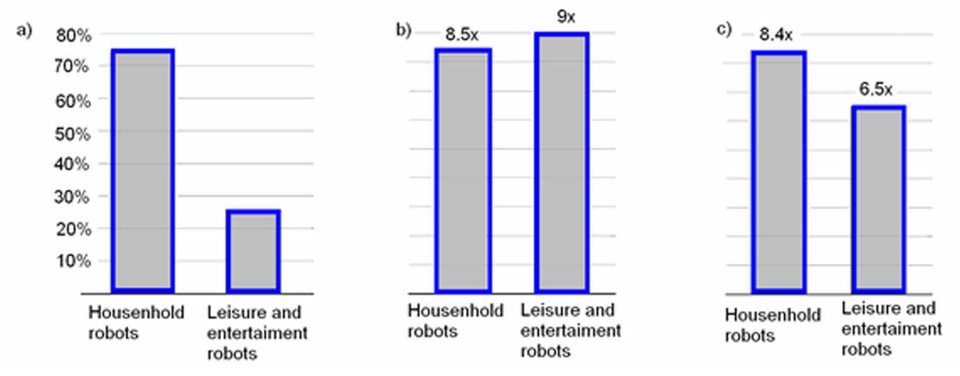
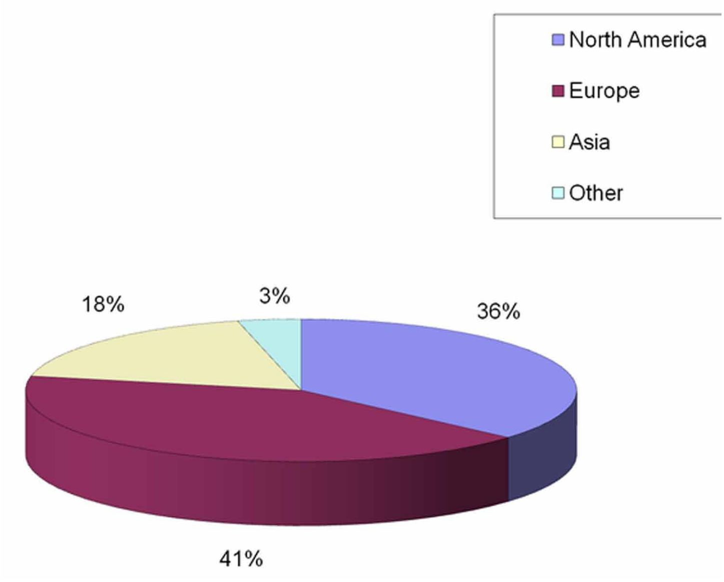


Figure 4. Number of service robots manufacturers (for all types of robots), diagram based on the IFR data (IFR Statistical Department, 2016).



Besides the study of the future market trends, an interesting issue is the prognosis of the service robot skills.

As it is presented in Table 1, currently used semi-autonomous robots will be replaced by skilled robots which act independently of humans.

Table 1. Service robots skills

Robot Category	Current Skills	Expected Skills in the Coming Years	Expected Skills in the Longer Perspective
Professional service providers	Semi-autonomous servicing robots (in that tele-manipulators and surgery robots)	Autonomous assistants	Fully skilled workers (in that autonomous medical caretakers)
Domestic service robots	Single task semi-autonomous devices	Autonomous domestic "tools"	Skilled home assistants (in that caretakers of children and elderly)
Security robots	UAV's, UGV's	Non-autonomous combat robots	Robotic squadrons (e.g. micro-robots, robots employing swarm intelligence)
Space robots	Semi-autonomous servicing robots	Autonomous servicing	Autonomous space exploration

DEVELOPMENT OF SERVICE ROBOTS

Beginnings

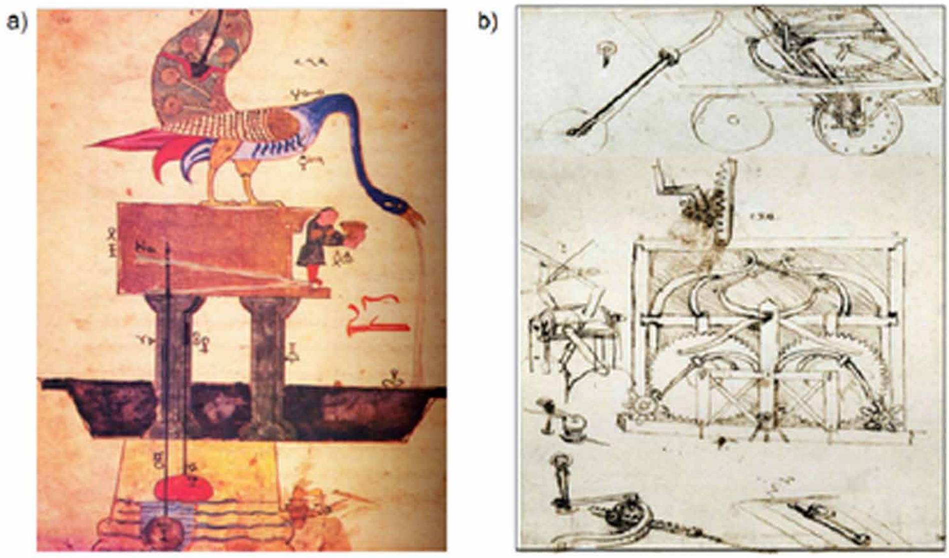
The idea of an artificial assistant performing some tasks appeared long time ago. According to the books of Iliad, written by Homer (8th c. BC), one of the Greek gods build different artificial servants. Some of them were human-like. In Ancient Egypt rope-operated statues of gods were used in the temples. In his *Politics* Aristotle stated that slavery will vanish when artificial tools will be introduced, performing autonomously useful work. This introduces the concept of service robot.

The physical roots of service robots can be traced to ancient "robotics" theatres. The engineering of theatre machines was undertaken in the antiquity in Greece and in the Roman Empire. Greeks reached the heights of knowledge even in the technical fields. An iconic example is the school of Alexandria, where since the 3rd century BC there was an intense activity in teaching and research, also pertaining to automatic devices. Hero of Alexandra (1st century AD), the author of: *Treatise on hydraulics, Treatise on pneumatics* and *Treatise on mechanics,* is named the precursor of entertainment robots, due to his theatres with moving figures (Rosheim, 1994).

Greek culture evolved and circulated combined with Roman technology. The Romans created some machines helping humans in their work. Marcus Vitruvius Pollio (born c. 80–70 BC), a Roman writer, architect and engineer, in his treatise *De Architectura* (On Architecture) described diverse machines helping humans in their work. Hoists, cranes and pulleys, as well as war machines such as catapults and ballistae, were described.

In the XII century AD Badi'as-Zaman Isma'il bin ar-Razzaz al-Jazari designed a hand washing automaton (Rosheim, 1994). It contained a male humanoid figure. When the user pushed a lever, the water drained and subsequently the figure refilled the basin. According to the sources al-Jazari's *peacock fountain* was a very sophisticated hand washing device with automatic servant offering soap and towel (Figure 5a). When pulling a plug on the peacock's tail the water was released out of the beak, the dirty water from the basin filled the bottom container rising a float, which pushed a linkage, which in turn made the servant figure come out from behind a door located bellow the peacock and offer soap. When more water was used a second float triggered the appearance of a second servant figure this time offering a towel. In his *Book of Knowledge of Ingenious Mechanical Devices* (1206) al-Jazari described fifty mechanical devices with instructions on how to build them. His famous "magic" water clock was a complicated mechanism decorated with many moving figures. Al-Jazari's works reveal an interest, not only in dramatic illusion, but in manipulating the environment for human comfort. He was fascinated by practical applications. This was the key element that was missing in the earlier Greek and Roman works. Therefore al-Jazari is attributed as one of the founding fathers of service robots.

Figure 5. Figure 2. Al-Jazari's peacock fountain (a). The original drawing of Leonardo's self-propelled cart (Codex Atlanticus by Leonardo da Vinci, circa 1478-80) (b).



An impressive example of intricate and skilful mechanical dolls are the mechanisms built in the XVIII c. by Swiss watch-makers: Pierre Jaquet-Droz, his son Henri Jaquet-Droz, Jean Frederic Leschat and Henri Millardet. Those dolls were programmable by exchanging pegs pushing cams. The dolls were capable of drawing, writing or playing piano with very precise motion of their fingers. As they were programmed, what was written, drawn or played could be changed.

Leonardo da Vinci's sketches (1495) of a mechanical knight, which can sit and move its arms and legs are considered as the first plan of humanoid robot. Da Vinci illustrated how a machine which imitates the anatomical structure of a human being can be constructed. Some sources claim that Leonardo presented the prototype of his mechanical knight in 1495 in Milano. Another famous design by da Vinci is the *self-propelled cart.* The mechanical structure was designed in such way that the system of springs enabled the alteration of the motion trajectory of the cart. The mechanism acted not only as a cart, but as a lever for the theatre's puppet (Figure 5b). Many other examples of service robot precursor can be recalled. Table 2 gives an overview of former achievements.

First Prototypes of Service Robots

It is difficult to give an exact year when the first service robots came to life, but it is obvious that the automatons recalled above can be treated as their remote precursors. Invention of electric motor at the end of the XIXc. was the first significant stimuli for robotics. The first robots resembling humanoids were built already at the beginning of XX c. for exhibitions and entertainment. They are closer precursors of service robots and from this point of view the history of service robots is older than the history of industrial robots which emerged in the 50ties of the XX century.

Probably one of the first more advanced prototypes was the humanoid Televox invented and patented by R.J.Wensley (USA) – Figure 8. Patent (submitted in 1927, published in 1930) covered supervisory control system applied in Televox, which was shown the in year 1927. The robot responded by simple actions to tuned tones.

In 1928 Model Engineers Society exhibited in London an electrically actuated humanoidal robot called Eric, designed by W.H.Richards – Figure 9.

The dog - Philidog (designed by M.Piraux from the Philips company) appeared during the Paris International Radio Exhibition (1929). The dog *"followed the movements of light, but when the lamp was put too close to its nose sensor ....it would become annoyed and start to bark!"*. For the 1939 New York World's Fair, a walking human-shape robot ELECTRO and his dog SPARKO were built. ELECTRO and SPARKO were designed by B.Barnett and produced by Westinghouse Electric Corporation – Figure 10. ELECTRO was capable of performing over 25 movements and responded to commands spoken into a microphone. SPARKO was the first

Table 2. Precursors of service robots

Period	Type of the device	Comment	Region/Inventor or Producer
V-I c. BC	Figures (god statues) with parts moved by a human	Figures with moving parts pulled by ropes	Egypt, Greece, Alexandria /persons serving the temples
III c. BC	Figures animated by water or steam water	The pneumatics laws were discovered and used for motion animation	Alexandria/Ctesibius
III c. BC	"Mechanical" orchestra. Figures were probably animated by ropes	Orchestra owned by the emperor Qin Shihunagdi. No details are known. Qin Shihuangdi was suffering from melancholy and, by the end of his life, ordered to destroy all books.	China/ not known
II c. BC	Pneumatically powered figures	Continuator of Ctesibius' work	Byzantium/Phylo of Byzantium
I c. AD	Theatres of moving figures	Roots of entertainment robots (Figure 6a) Designer of mechanical hammer Figure 6b	Alexandria/Hero of Alexandria
III c. AD	Walking machine Mu Niu Liu Ma (precursor of walking robots)	According to the descriptions it was an animal shaped wheelbarrow with legs transferred in a sequence observed in horse slow walk. It was used for transportation.	The construction was supervised by Zhu Ge-Liang/China
XI c. AD	Su Song Tower	Clock Tower with water driven escapement (Figure 7)	China/ Su Song
XII c. AD	Early "programmable" automatons, in that "robotic" hand	First complicated mechanical devices with many moving parts	Turkey/Badi'as-Zaman Isma'il bin ar-Razzaz al-Jazari
XV-XVI c. AD	Moving platforms carrying figures	Self-propelled cart (Figure 5b)	Italy/Leanardo da Vinci
XVI-XVII c AD	Dolls dancing and/or playing diverse musical instruments (automatons)	Hans Bullmann (Germany) is often attributed being the first creator of an android in a human form (1525)	Italy, Germany, Austria/ Juanelo Torreano, Turriano from Cremone, Hans Bullmann, Christoph Margraf
XVIIc. AD	Diverse automatons	Complex automations, in that machine animating a whole fighting army	France/ Christiaan Huygens
XVIII c. AD	Digesting duck, flute player imitating the birding	First automatons exhibiting the functions of a biological organisms	France/ Jacques de Vaucanson
XVIII c. AD	Programmable automatons; dolls writing different words, playing piano	The piano playing doll hand moved its fingers very fast, the doll played music like a real human	Pierre Jaquet-Droz, and his son Henri Jaquet-Droz/Switzerland
XVII- XIX c. AD	Steam engines with legs (precursors of walking robots)	Probably the first walking devices with on-board actuation	England/T.Brunton, D.Gordon
XIX c. AD	Feet Walking Machine (Stopochodjaszczaja Machina)	The proportions where carefully evaluated, the leg end transfer sequence and leg-end trajectories were such as observed in animals.	Russia/P.L.Chebyshev
XIX c. AD	Karakuri: mechanical toys serving tea, shooting arrows or painting		Japan/Hishashige Tanaka

Figure 6. Selected works of Hero from Alexandria: Hercules and a dragon: reconstruction of one of many Hero's automatons made by Giovanni Battista Aleotti in 1589 (a), mechanical hammer (b).

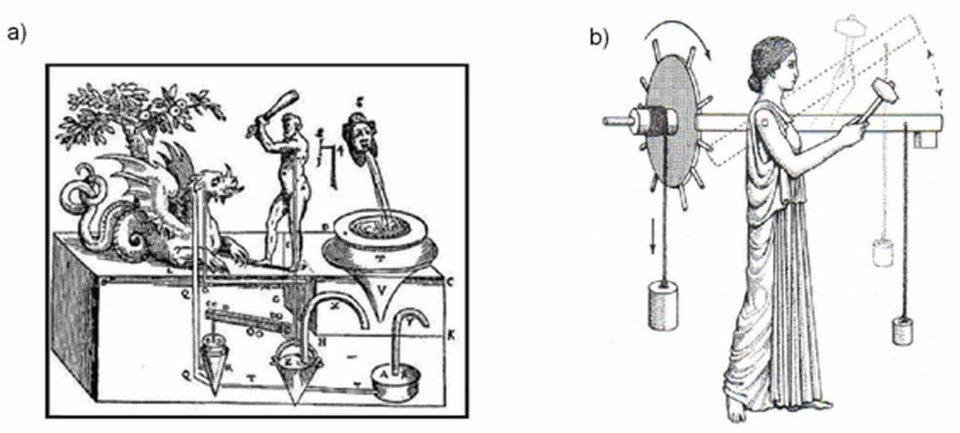


Figure 7. The copy from Su Song's work showing the clock tower. His clock used the endless power transmitting chain - the "celestial ladder". Su Song applied it in his clock tower, and described in his book Xinyi Xiangfayao (1092).

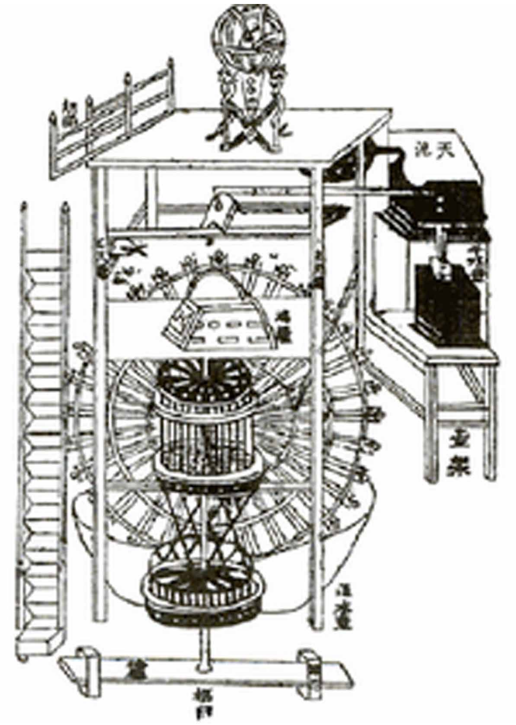


Figure 8. Televox robot with its constructor R.J.Wensley (a), the supervisory control system –submitted for patenting - drawing by R.J.Wensley.

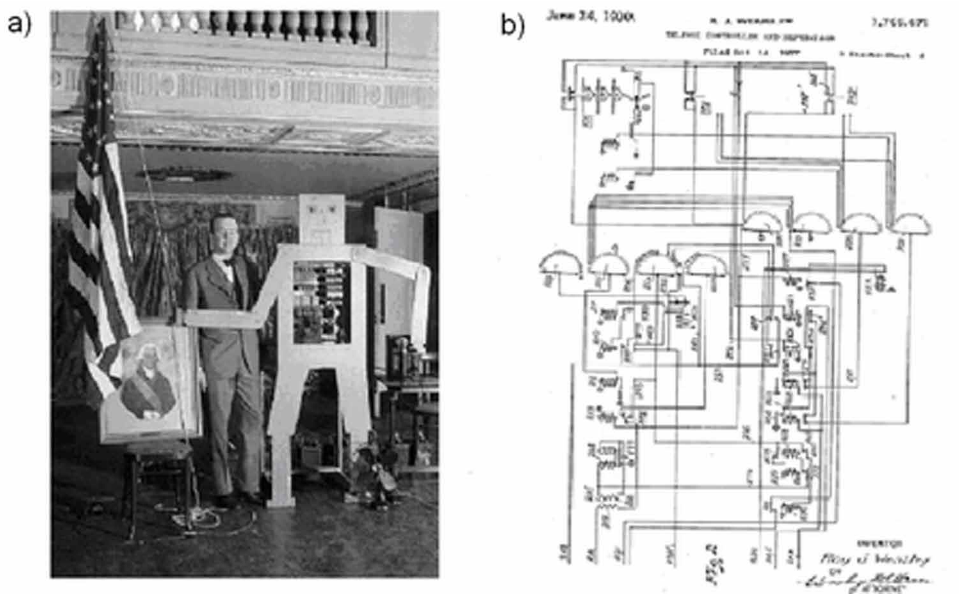
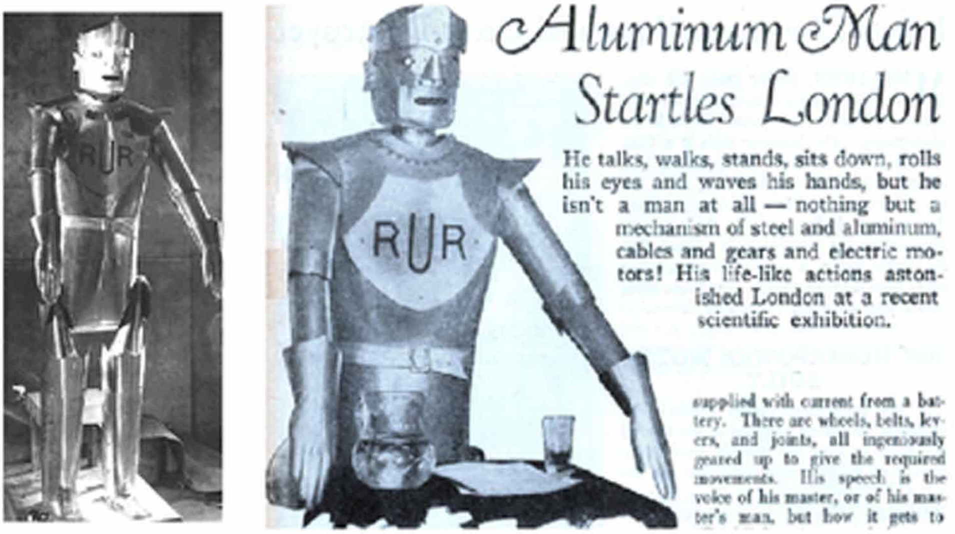


Figure 9. Robot named Eric RUR (RUR comes from Carel Capek's play RUR – Rossum's Universal Robot, where the term robot was coined) – the robot photograph and the title page of the article /Chronicle Telegram 26 Nov 1928/.

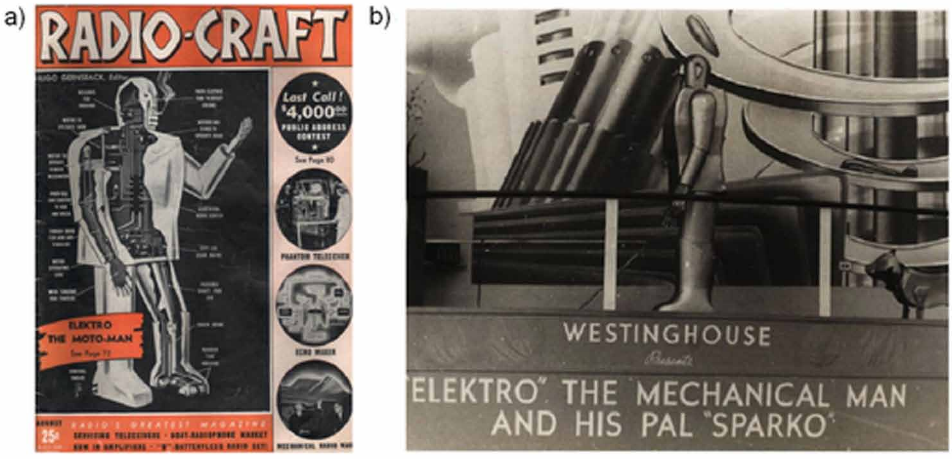


Aluminum Man Startles London

He talks, walks, stands, sits down, rolls his eyes and waves his hands, but he isn't a man at all — nothing but a mechanism of steel and aluminum, cables and gears and electric motors! His life-like actions astonished London at a recent scientific exhibition.

supplied with current from a battery. There are wheels, belts, levers, and joints, all ingeniously geared up to give the required movements. His speech is the voice of his master, or of his master's man, but how it gets to

Figure 10. The cover page of Radio-Craft magazine (1939) describing ELECTRO robot (a), the poster from New York World's Fair advertising ELECTRO and SPARKO.



robotic victim of the car accident. Just before the exhibition opening, SPARKO went out of the building attracted by the headlights of an automobile and was destroyed.

In 1948 first robots with biological behaviours - Elsie and Elmer, were created by William Grey Walter. Those tortoises could dance with each other due to their attraction to light sources. In this case the bulbs were attached to each robot. The resultant effect was obtained without any specific perception of the other robot and with no special social rules implemented .

In general XX c. was marked by an extensive development of diverse robots, especially in its second half, due to the development of computerized control methods.

In the year 1962 in Russia a mobile robot named Sepulka was elaborated. It was designed as a tour guide – Figure 11. The robot started its work in museum in the year 1963 and in the year 2010 was still active. It was probably the first robot designed for that purpose.

In the years 1966-1972 the Artificial Intelligence Centre at Stanford Research Institute developed a mobile robot named Shakey – Figure 12. The robot had limited perception, but was able to plan its route and rearrange objects in its surroundings. The work on this project brought progress to artificial intelligence methods. In 1972 the robot was referred to by the Life Magazine as the first *electronic person*.

Thirty years after creating in 1954 Unimation - the first company producing industrial robots, J.Engelberger founded in 1984 Transitions Research Corporation which became HelpMate Robotics Inc. Since its beginnings this corporation has focused on mobile robots equipped with sensors, servicing humans and especially the robots for elderly care and for medical applications.

Figure 11. Russian tour guide robot Sepulka (Management of the Polytechnic Museum permits the use of the SEPULKA robot images).

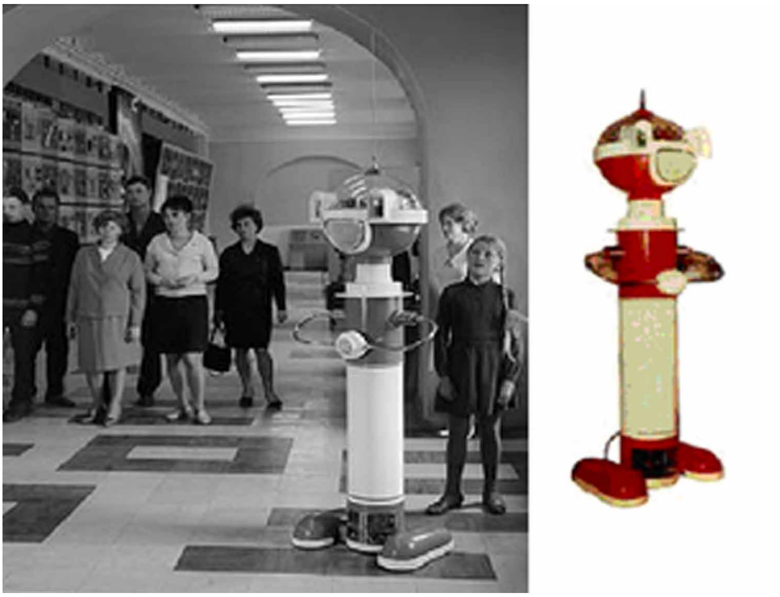
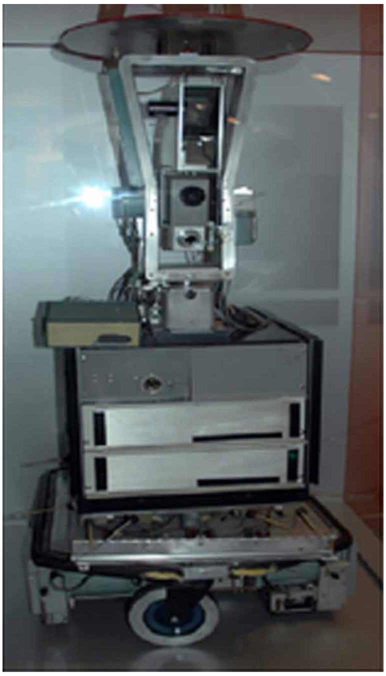


Figure 12. Robot Shakey built by Artificial Intelligence Centre at Stanford Research Institute (public resources).



In 1987 Hughes Research Laboratories demonstrated autonomous mobile robot navigating in cross-country environment. In the years 1990-1994 Carnegie Mellon University developed Dante I and Dante II – robots for volcano exploration. In the year 1997 NASA sent the mobile robot Sojourner to Mars. The robot performed its mission successfully.

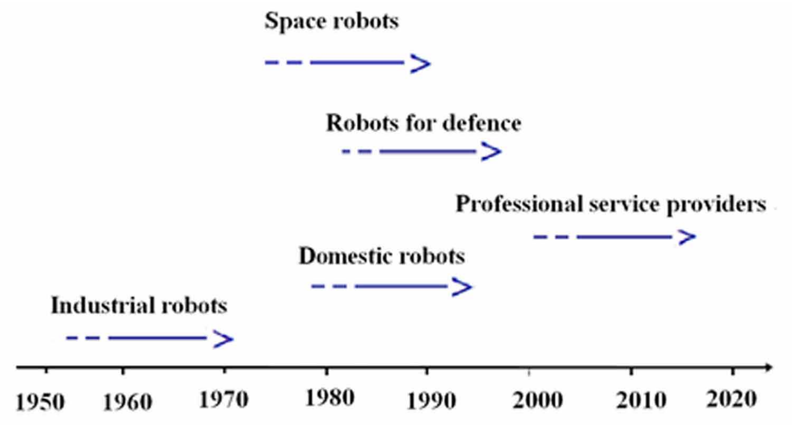
The tour guide Minerva created in the USA (1997) had learning capabilities and was able to interact with people. It was endowed with simple emotions expressed by smiling or signing.

By the mid 90ties the Massachusetts Institute of Technology elaborated the robot Cog consisting of an upper torso of a humanoid robot. Cog was to imitate human-like intelligence. Later the same laboratory designed Kismet, an emotional robot with an articulated face Kismet simulates emotions through facial expression, vocalisation and movement.

Service robots were also developed for the jobs, where task execution by a human is dangerous, impossible or unacceptable. Those robots were created as: mobile platforms, manipulators, tele-manipulators or mobile robots with manipulation abilities.

As the survey of robotics literature shows, the first helpmate indoor mobile robots appeared in the 80'ties of the XX c. Fast progress in sensing technologies, development of control equipment, improvements of actuators and power supplies, which occurred in the 80ies and early 90ies of the XX c. resulted in many robot projects dedicated to new non-manufacturing applications. Figure 13 illustrates the milestones in robot development. Service robotics research on space and domestic robots came first. From the beginning of the XXI c. the stress is laid on professional service provides.

Figure 13. Service robot development.



Recent Challenges

The recent years are marked by a very fast development of different service robots. Their abilities increased significantly. Staring from the Waseda piano playing robots WABOT-1 (1973), WABOT-2 (1984) – Figure 14, Waseda flutist robot (WR-1 - 1990, WR-4RIV - 2009) we come to the robots participating in sport games or dancing together with humans. To this group belongs Topio (**TO**SY **Pi**ng Pong Playing Rob**o**t) developed by robotic company TOSY in Vietnam (2007) – Figure 15. The robot improves its skills by activly learning when playing with a human. One of the most well know humanoids is the robot ASIMO, developed by the Honda company. Honda started the work on humanoids in the 80ties of the XX c. displaying in 1986 their first two-legged robot E0. ASIMO was presented for the first time in the year 2000 and since that time has undergone continuous evolution in size, weight and improved quality of motion as well as expanded human-robot interaction – Figure 16.

Humanoid HRP-4C demonstrated in 2009 created by the Japanese National Institute of Advanced Industrial Science and Technology sings and dances with human dancers. This robot mimics human emotions - it is considered to be a gynoid (*gynoid* from Greek is anything what resembles a woman - Figure 17.

In the year 2010 NASA and General Motors demonstrated the Robonaut 2, an advanced humanoid for space walking. In February 2011 the robot was sent on a space mission.

Figure 14. WABOT-2 - piano playing robot WABOT-2©[2012] Humanoid Robotics Institute, Waseda Universit. (Used with permission)

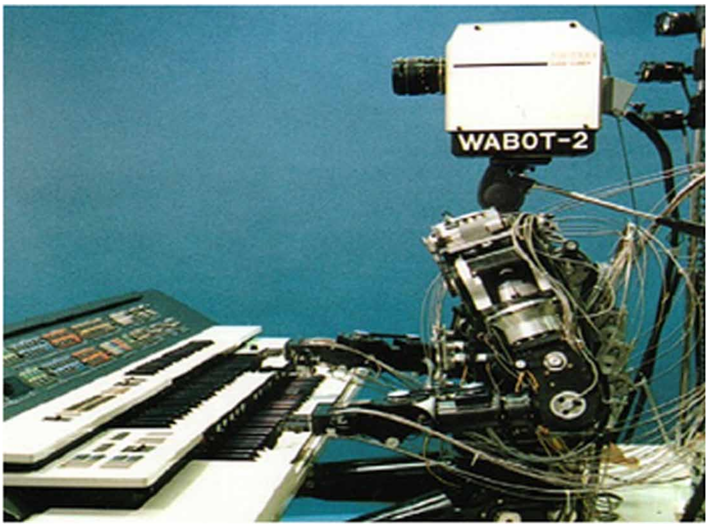


Figure 15. Topio - Vietnamese ping pong playing robot© [2012], TOSY Robotics JS Company, Vietnam (Used with permission)

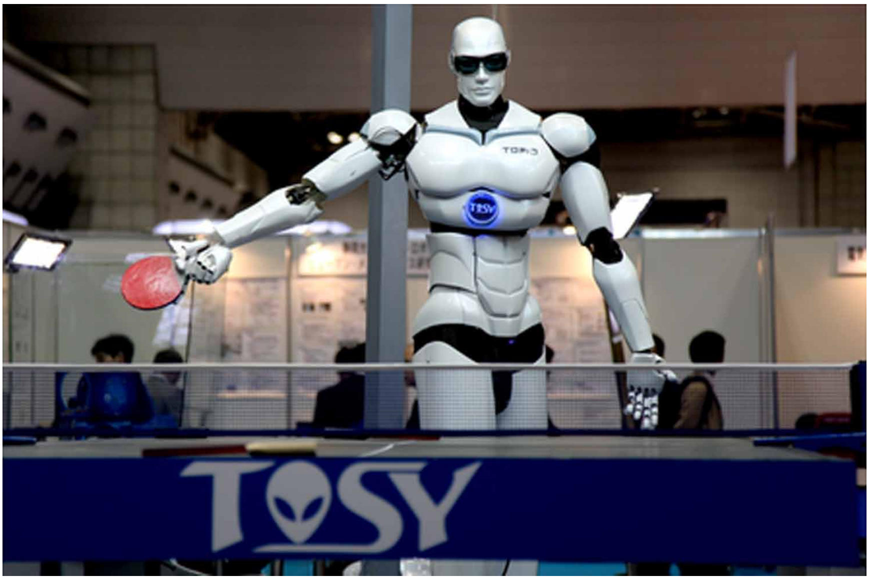


Figure 16. Robot ASIMO (photo by the author T.Zielinska).

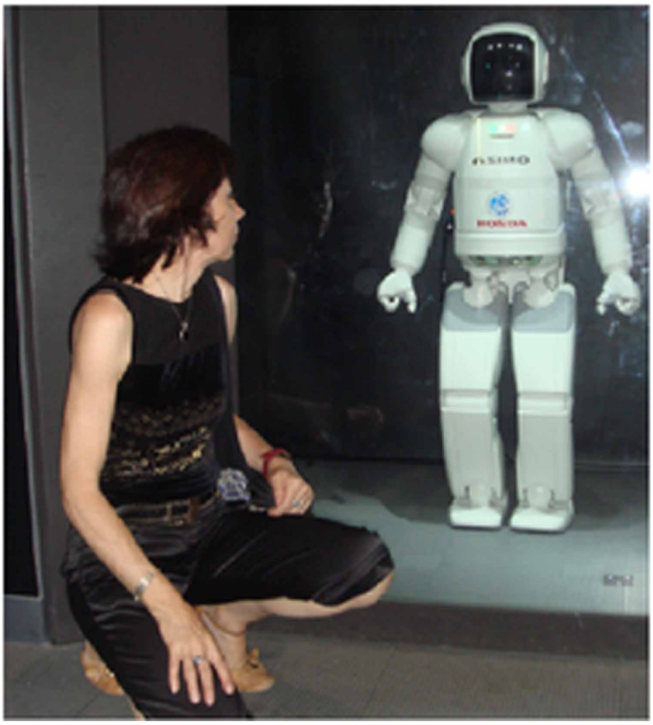
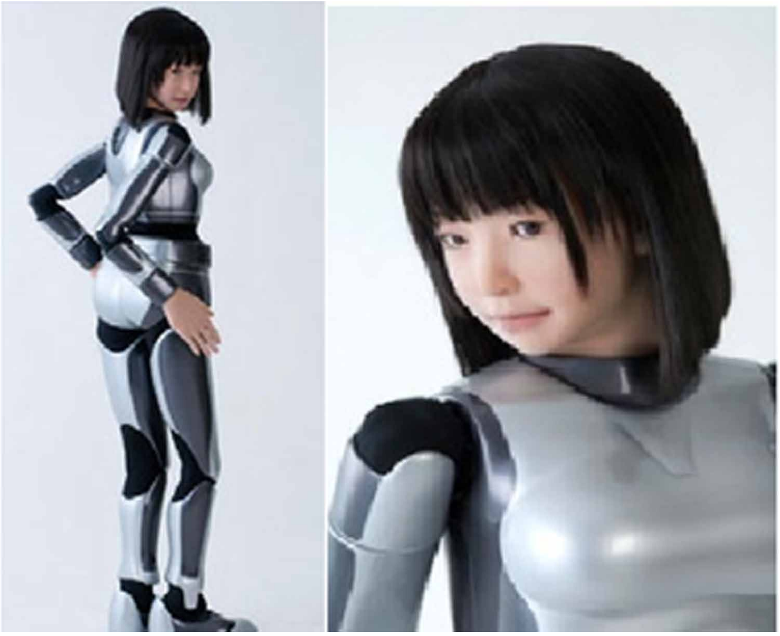


Figure 17. Gynoid - humanoid HRP-4C – cAIST (http://www.aist.go,jp)© [2012],National Institute of Advanced Industrial Science and Technology, Japan (Used with permission)



UAVs and AUVs form a specific type of service robots. The precursors of UAVs (Unmanned Aerial Vehicles) are the flying autonomous bombs (V-1, V-2) used during the Second World War. Till our days a variety of UAV's of different sizes, abilities and ranges has been designed – Figure 18. They have been used for military and civilian missions, such as search and rescue operations as well as for monitoring the environment, or checking the state of pipelines .

Autonomous Underwater Vehicles (AUVs) known also as Unmanned Undersea Vehicles (UUVs) are robots acting in water – Figure 19. They are used to make maps of the seafloor, to monitor underwater pipelines or to perform military or research missions.

Surveillance, patrol and security mobile robots form a large group of service robots. Military robots, similarly to the UAVs, can have their roots traced to the Second World War. Goliaths, the remotely controlled tracked mines (Germany) or tele-tanks (Russia). This type robots are named UGVs (Unmanned Ground Vehicles). Currently they are autonomous and equipped with diverse sensors, tools or are can even be armed (Figure 20).

Figure 18. Chameleon - the first Polish mini-UAV tested under field conditions in 1995 ©[2012] C.Galinski (Used with permission)

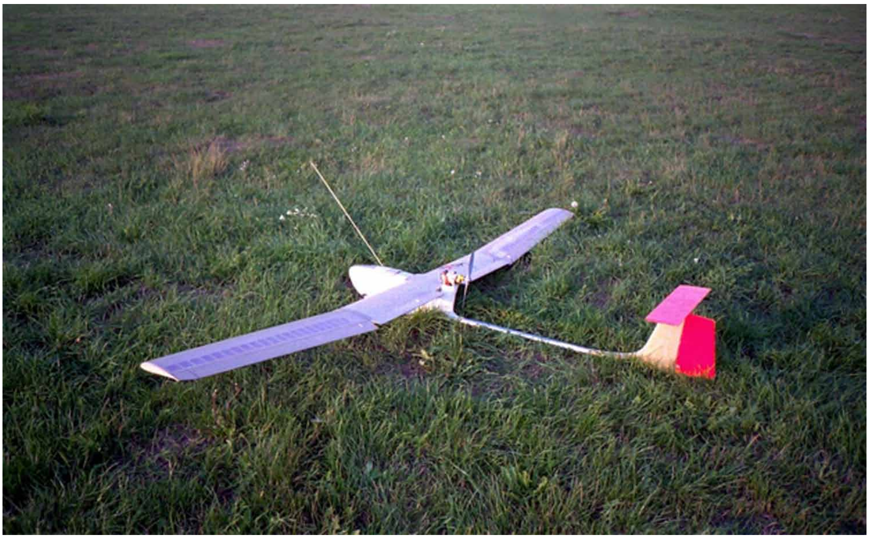


Figure 19. AUV produced by Bluefin Robotics Corporation ©[2012] Bluefin Robotics (Used with permission)

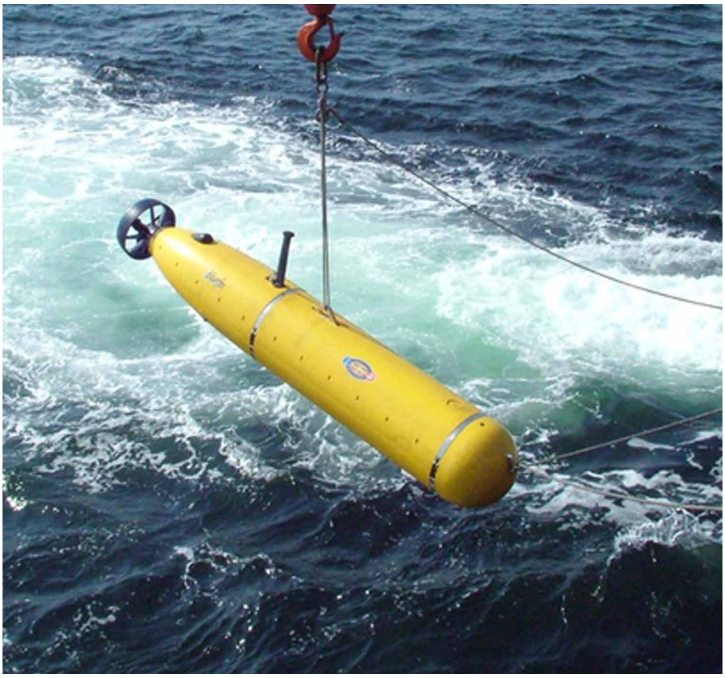
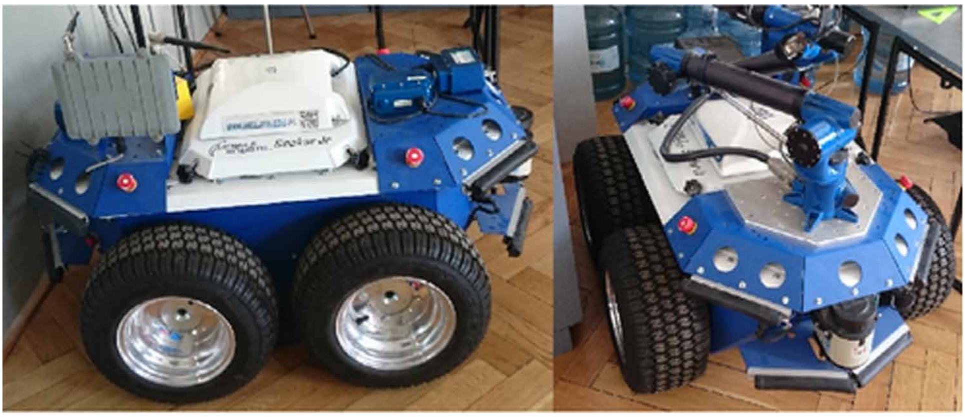
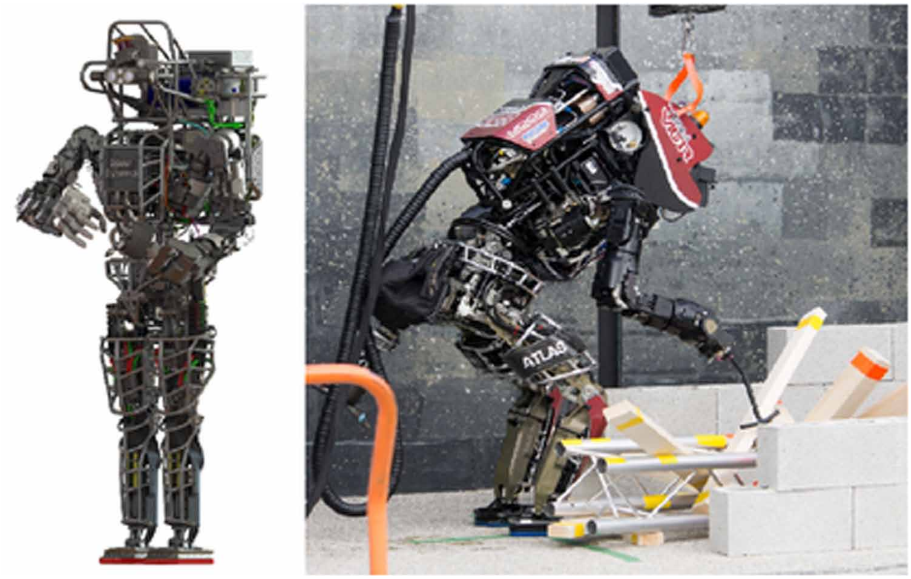


Figure 20. UGV Seekur Jr. – autonomous outdoor robot (photo by T.Zielinska).



The most impressive achievements to a large extent are stimulated by military applications. In the year 2013 the Atlas robot, developed for the Defense Advanced Research Projects Agency *(*DARPA*)* by Boston Dynamics, was considered as the most advanced humanoid of the World – Figure 21. It can use tools, is able to locate objects using its cameras and laser range finders. The robot can walk and crawl. It was built to take part in rescue operations. Its older "brother" Petman (**P**rotection

Figure 21. Humanoid Atlas built by Boston Dynamics for DARPA (from DARPA public resources)



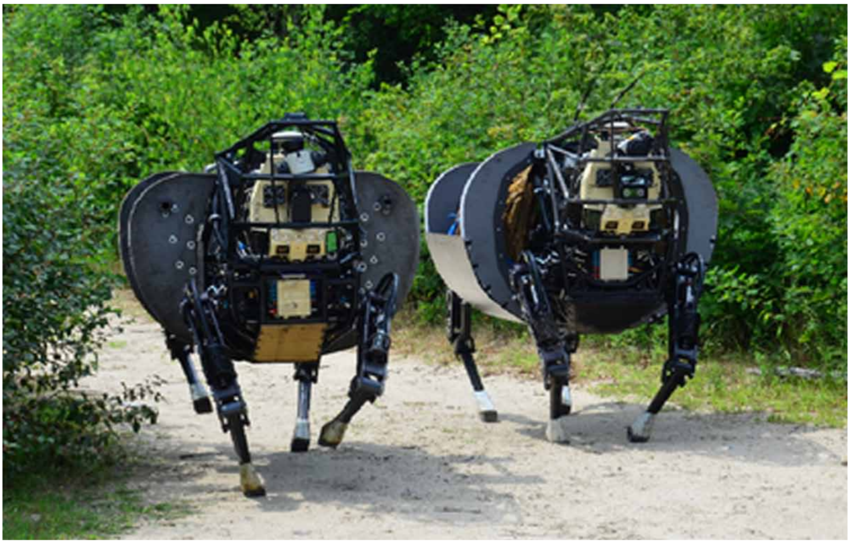
Ensemble **T**est **Man**nequin), also developed by Boston Dynamics for the U.S. Army, is used to test the special soldiers' protection clothing. PETMAN can perform suit-stressing exercising, and other complex movements enabling testing of chemical exposure on the suits. Both robots imitate closely the human body structure and movement properties.

Other advanced robots for military applications are the Boston Dynamics four-legged robots also developed for DARPA. The robots were demonstrated for the first time in 2012. They composed the so called Legged Squad Support System (LS3) for supporting marines – Figure 22. The robots can move semi-autonomously transporting heavy loads in demanding terrain.

In 2014 DARPA started to implement on an industrial scale the outputs of the programs called Adaptive Vehicle Make (AVM). The aim of those programs was to develop new a approach to the design, testing and manufacturing of complex military systems, what should significantly decrease their development time (DARPA, 2014).

The robots designed for rescue operations, the robots for humanitarian demining and robots for space exploration act in hazardous environments. The idea of using mechanical devices and vehicles for demining operations was conceived during the First World War (Havlik, 2008). The robotic demining technology is applied in mine searching and neutralization. Those are specially designed wheeled robots or robots on tracks, however there are also attempts to apply walking machines. A demining robot needs a special mechanical design assuring the demanded mobility.

Figure 22. Legged Squad Support System built by Boston Dynamics for DARPA (from DARPA public resources).



They use high quality general purpose and dedicated sensory systems with special, very reliable, data fusion methods. The vehicles must be able to act with partial or full autonomy. Mine-seeking robots are at the early developmental stage, they still lack needed flexibility and they are still too costly (Maki, 2008).

Robot companions are robots designed for personal use at home. They should be able to perform a variety of tasks, such as helping in education, guarding, cleaning, cooking, entertainment, shopping and message delivery. They can have the different shapes, such as a robot pet, robot humanoid, mobile platform with manipulators or other. According to (Dautenhahn, 2007): "*a robot companion is a robot that: - makes itself 'useful', i.e. is able to carry out a variety of tasks in order to assist humans, e.g. in a domestic home environment, and - behaves socially, i.e. possesses social skills in order to be able to interact with people in a socially acceptable manner"*. Current robots are still rather far from the abilities that were expected.

In the year 2005 Mitsubishi Heavy Industries introduced the first Japanese domestic robot Wakamaru, to become a companion to the elderly and disabled people.

The open community Robot Companions for Citizens established in the year 2010 consists of scientists who promote the robot companion program as one of most important research priorities for the nearest future.

Service robots are also more extensively applied in the medical field. The first robots for medical applications were derived from industrial robots. In 1985 the PUMA 560 was used to perform neurosurgical biopsies under computed tomography guidance CITE. First medical robots were mainly applied in innovative surgery and in rehabilitation. Now they support a wide scope of treatments resulting in less invasive surgery. In the roadmap of robotics (SPARC 2016) providing a strategic overview of European robotics the following groups of currently available medical robots were distinguished (from the point of view of the application domains): clinical robots (for surgery, for diagnostics, for assisting medical treatment), rehabilitation robots (in that also exoskeletons, prosthesis, orthoses), robots for special purposes and assistance (robots helping to perform routine functions).

One of the most well-known surgical robots is the da Vinci robot. The data collected in the year 2005 indicated that more than 800 hospitals in the United States and Europe used the da Vinci robot in various types of surgery. This number has increased ever since. Da Vinci system is undergoing continuous improvements and till now has been successfully applied in general surgery, cardiothoracic surgery, urology, gynaecology, and in otolaryngology (Rajesh Aggraval et all. 2014). The group of more advanced medical service robots is represented by - the voice-controlled robots serving patients food, drinks and medicines, - the diagnostics robots - intelligent robotic wheelchairs, - robot companions predicting the human actions and interacting with the elderly or with the patients with motion problems. Very helpful are robots controlled by brain generated electric signals (Electroencephalographic signals –

EEG) or by electric activity of the muscles (Electromyography signals – EMG). EMG controlled robotic suits can provide their wearer with supernatural powers, in the medical field they are used as a powerful prosthesis or as a human controlled servicing devices, EEG controlled robots are used to fulfil the commands of a paralysed persons. A special group of medical robots is composed of the "social" robots, interacting with humans on the emotional level. They are applied to support the treatment of autistics children or to interact with the old patients with mental problems. Paro is here an example. It is a baby seal developed for patients suffering from dementia. Created at the National Institute of Advanced Industrial Science and Technology (AIST) in Japan, Paro is equipped with microphones and several sensors detecting light, sound and the other stimuli. The robot interacts with patients through movements of its fins and eyes. In general the elderly care robots are split into two separate groups: rehabilitation-oriented and socially-oriented robots. It is difficult to satisfy rehabilitation and social interaction purposes in one device, but Paro is a special case catering to both. It has a stimulating as well as a calming effect on the patients, facilitating discussions and interactions among dementia patients. Paro represents the first commercially available care robot dedicated to dementia patients. In the future in the medical field mico-robots and nano-robots will be used. The examples are: robotic *pills* delivering the required doses of medication at the required time, mico-robots travelling though the blood vessels performing diagnostic functions or the robots inducing some medical treatment inside the body. In the year 2016 researchers from MIT, the University of Sheffield, and the Tokyo Institute of Technology demonstrated a miniature robot that can unfold itself from a swallowed capsule. The robot steered by external magnetic fields, can crawl on the stomach wall removing a swallowed button battery or patching a wound (Hardesty 2016). There are several projects devoted to robotic *pills* (ACS 2015). According to (SPARC 2016) the priority in medical robotics is the integration of bio-sensing into the sensing system of a robot and the development of such robotic systems that minimise the intrusiveness of medical sensing.

Another growing fast service robot application area is agriculture. In agriculture and forestry, research into driverless vehicles was initiated in the early 1960's of the XX c. with projects on automatically steered systems and autonomous tractors. In the 80s there were first attempts to apply robotic harvesting to horticultural products and robots picking fruits from the trees, mainly citruses. But the economic motivation decreased in the 90s due to the availability of cheap human labor. Currently the application of robotic systems in agriculture is experiencing an increased interest. The use of robots in agriculture creates many ambitious research goals - harvesting needs manipulation of the objects that can be easily damaged, agricultural products are of different sizes, shapes, having different material properties, so they must be harvested in different ways. There are attempts to develop adaptable vehicles using diverse behavioral control methods. A combined application of complex sensory systems, communication technologies, and global positioning systems (GPS) with geographical information systems (GIS), for navigation, help the development of autonomous vehicles dedicated to harvesting, for horticulture sector, and for landscape management (Pedersen et all. 2008). In recent years special robots are also developed to help farmers in weed control and planting certain types of crops (Aminzadeh, 2014). In general we can distinguish robots for milking, mobile autonomous barn cleaning robots, mobile autonomous feeding robots, farming and horticulture robots (planting, pruning and harvesting) and robots for other purposes (landscape management, weed control etc.).

It must be noted that in the case of assistive and service robots quite often the requirements imposed on the robot by the user are virtually unlimited while the capabilities of the robot on-board control computer are limited. The solution of this dilemma is through the utilization of resources located in a cloud. Upon formulation of the task by the owner of the robot vocally, its controller contacts the cloud resources for a module that solves the problem. However this requires the robot control system to have not only a variable structure, but also switchable supervisory responsibilities, as only the task executing module knows what the robot should do. Such a system was designed within a 7FP EU project RAPP (Robotic Applications for Delivering Smart User Empowering Applications - Zielinski et all. 2017). The secure networking, allows robotic clouds to access a huge amount of data. A serious technical problem that researchers meet when develloping the robotics clouds is the system integration when the components use communication standard, this topic was a subject of several big research projects as well.

From the above overview one can conclude the nowadays service robotics is going through a major transformation. Intensive deployment is expected of novel types of robots and drones in new civilian applications, in that especially in agriculture and forestry. Historically there has been significant interest in forestry robotics within Europe (SPARC 2016). The felling of trees is now highly automatized, but the use of robotics technology to monitor forests is still in its infancy. This is an emerging domain for application of drones, however current regulations are a barrier.

The networked and service robots which are often user interfaced, such as rescue robots, human assisting robots, health care robots and robots for military applications, have been identified by the USA National Institute of Standards and Technology (in collaboration with Department of Homeland Security) as a class of devices which require concurrent design of hardware and software including the security mechanisms (Dutta, 2015). It means that the control software must be selected taking into account the specifics of hardware, mechanical and sensory parts. The faults serving mechanisms must be developed as well and the whole system must assure a proper level of active safety. The software elements detecting malfunctions, failures, and faults must be implemented together with mitigation strategies. An adequate modular functional architecure of the control software is the necessity in complex autonomous robots (Zielinska, 2006).

The gate to the widespread application of service robots was opened wide with the introduction in February 2014 of the new ISO 13482 Safety Standard for mobile servant robots, physical assistant robots and person carrier robot (BS EN ISO 13482, 2014). This is the first safety norm issued by the International Organization for Standardization (ISO) which allows robots and humans to interact directly and share the same space. Existence of such a standard is essential for enabling the commercialisation of personal robots.

PERSPECTIVES

The further development of robot intelligence is associated with cloud robotics, freeing robots from their computational limits and thus increasing their capabilities. This will enable the development of sensor rich robots with high adaptability and autonomy. The deep learning algorithms will allow the robots to access and process efficiently the data available in the media, by that increasing their perception abilities, what will result in significant enhancement of their “understanding”. The robots will learn faster and more efficiently. Knowledge sharing between robots is another new trend requiring a common ontology. Very different robots having different sensors and different effectors will share their “knowledge”, thus improving their learning processes.

Looking at the robot design philosophy the most popular recent trend is biological inspiration, thus biologically inspired robots emerged (Laschi et all. 2016). The soft and reconfigurable robots very often imitate biological species, and are designed having in mind very specific needs of human servicing, where soft devices can make much less harm during the system fault or failure (Rossiter et all. 2016). For systems of such type bio-inspired morphological computations are often used (Mintchev at all, 2016). Here the system behaviour is not seen as the result of perception, processing and acting, as it is usual, but depends significantly on the mechanical properties - the change of the robot shape (the so called morphological adaptation), and proper arrangement of perception, actuation and processing units. The material properties are here very important. In reconfigurable, and/or soft robots the typical model-based approaches become insufficient. Morphological computation focuses on the role of soft bodies deformation and uses reaction forces to obtaining the needed behaviours. By this the control complexity is reduced, but the control philosophy is very different than before. The action is obtained not only by producing the actuating signals, but by passive and active compliance and change of the body shape. The analogy to soft bodied animals and plants is here often used.

Considering human – robot interaction the future belongs to the robots exhibiting social intelligence and recognizing not only the human emotions, but also motion intentions. The robots will be not only capable of performing voice communication, gesture and emotions recognition, as well as human intention prediction, but they will be able to realise complex actions interpreting human brain EEG (Electroencephalography) signals. First results obtained in this area are very promising (Millán et all, 2010), (Velasco-Alvarez et all, 2013).

Social and physical interactive skills are necessary in many applications where robots interact and collaborate with other robots or humans. Sensorimotor skills, including locomotion, object manipulation, carrying loads etc., are fundamental for robot companions or robot workers. According to (IFR Statistical Department, 2010-2016) the stress will be put on providing the support for the elderly people. Such tasks need complete reliability of operation, safety and full autonomy. Those features are expected also from military robots, which strongly stimulate the progress in service robotics.

Figure 23 presents developmental trends in service robotics. As it is illustrated, the first robots were teleoperated devices, while currently fully autonomous, self-deciding robotic teams are rapidly becoming a reality. Figure 24 gives an overview of robotic technologies. The history of industrial robotics started with manipulators

Figure 23. Development of service robots (according to Christensen, 2004, updated by the author 2017).

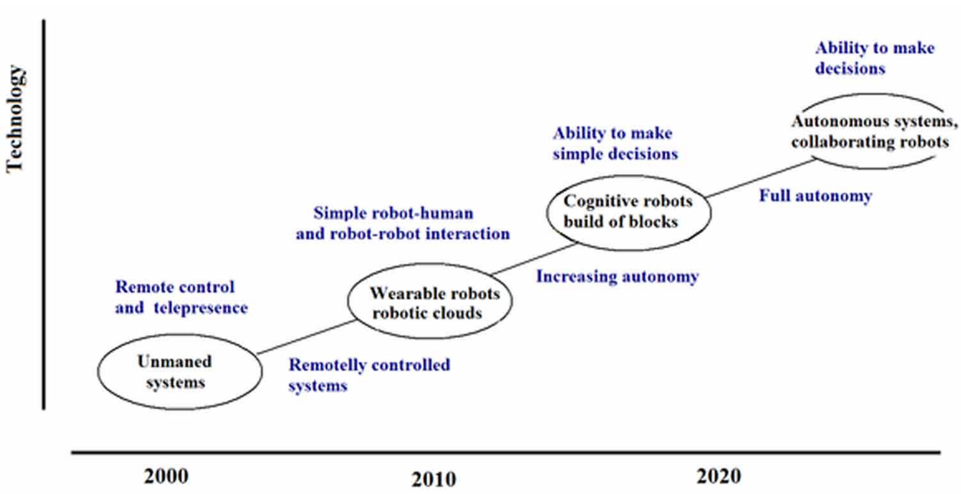
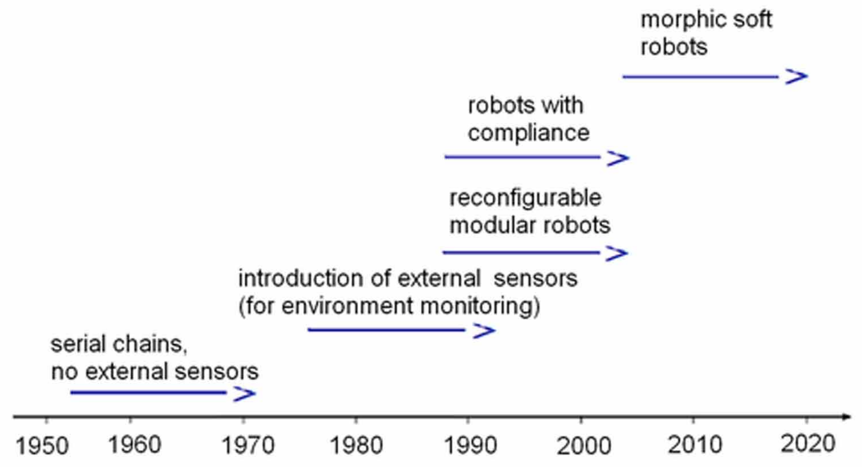

Technology
Ability to make decisions
Autonomous systems, collaborating robots
Ability to make simple decisions
Full autonomy
Cognitive robots build of blocks
Simple robot-human and robot-robot interaction
Increasing autonomy
Remote control and telepresence
Wearable robots robotic clouds
Unmaned systems
Remotelly controlled systems
2000
2010
2020

Figure 24. Robotic technologies.



constructed of rigid links, while the future robots will be made of soft materials, some of those robots having reconfigurable structure.

In the VIII century B.C., in the 18th book of Iliad Homer described maidservants that were built by Vulcanos to serve Gods: they were propelled by wheels, nicely human shaped, and were able to speak, exhibiting some intelligence. Several other impressive machines described by Homer were automatically operated. More than two thousand years later we live in the world where such fantasies are becoming real. The future is open to personal robots directly interacting with humans and being the skilful helpers or companions.

REFERENCES

ACS - American Chemical Society News Service Weekly Press. (2015, June 17). *Toward nanorobots that swim through blood to deliver drugs.* Author.

Aggraval, Darzi, & Yang. (2014). Medical Sciences part II, Robotics in Sugrery - Past, Present and Future. In *Encyclopedia of Life Support Systems*. Retrieved from www.eolss.net

Aminzadeh, R., & Fotouhi, R. (2014). Novel design of a precision planter for a robotic assistaint farmer. *Proc. of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE.*

BS EN ISO 13482:2014 (2014). Robots and robotic devices. Safety requirements for personal care robots. British Standards Institution.

Ceccarelli, M. (1998). Mechanisms Schemes in Teaching: A Historical Overview. *Journal of Mechanical Design*, *120*(4), 533–541. doi:10.1115/1.2829311

Ceccarelli, M. (2001). A Historical Perspective of Robotics Towards the Future. *J. of Robotics and Mechatronics*, *13*(3), 299–313. doi:10.20965/jrm.2001.p0299

Christensen, H. (2004). *European Service Robotics. A white paper on the status and opportunities of European Service Robots*. EURON-IFR.

DARPA. (2014). Retrieved from www.darpa.mil

Dautenhahn, K. (2007). *Socially intelligent robots: Dimensions of human–robot interaction.* Philosophical Transactions. *Royal Society.*, *362*(1480), 679–704. PMID:17301026

Dutta, V., & Zielinska, T. (2015). Networking technologies for robotic applications. In *International Conference on Cyber Security (ICCS).* California State University. Retrieved from http://iccs2015.iaasse.org/

Engelberger, J. (1989). *Robotics in Service*. MIT Press. doi:10.1007/978-94-009-1099-7

Habib, M. K. (2008). *Humanitarian Demining: the Problem, Difficulties, Priorities, Demining Technology and the Challenge for Robotic*s. *I-Tech Education and Publishing*.

Hardesty, L. (2016). *Ingestible origami robot*. MIT News Office.

Havlik, S. (2008). Land Robotic Vehicles for Demining. *I-Tech Education and Publishing*. doi:10.5772/5419

IFR Statistical Department. (2013). *Executive Summary.* World Robotics 2013: Service Robots.

IFR Statistical Department. (2014). *Executive Summary.* World Robotics 2014: Service Robots.

IFR Statistical Department. (2016). *Executive Summary.* World Robotics 2016: Service Robots.

Laschi, C. I., & Mazzolai, B. (2016). Lessons from Animals and Plants. IEEE Robotics and Automation, 23(3), 107-114.

Millán, J. D. R., Rupp, R., Müller-Putz, G. R., Murray-Smith, R., Giugliemma, C., Tangermann, M., ... Mattia, D. (2010). Combining Brain–Computer Interfaces and Assistive Technologies: State-of-the-Art and Challenges. *Frontiers in Neuroscience*, *4*, 161. PMID:20877434

Mintchev, S., & Floreano, D. (2016). Adaptive Motphology. *IEEE Robotics and Automation*, *23*(3), 42–54. doi:10.1109/MRA.2016.2580593

Pedersen, S. M., Fountas, S., & Blackmore, S. (2008). *Agricultural Robots – Applications and Economic Perspectives*. Retrieved from www.intechopen.com

Pransky, J. (1996). Service robots – how should we define them? *Service Robot. International Journal (Toronto, Ont.)*, *2*(1), 4–5.

Rosheim, M. E. (1994). *Robot Evolution: The Development of Anthrobotics*. New York: John Wiley.

Rossiter, J., & Hauser, H. (2016). Soft Robotics - The Next Industrial Revolution? *IEEE Robotics and Automation*, *23*(3), 17–20. doi:10.1109/MRA.2016.2588018

SPARC The Partnership for Robotics in Europe. (2016). *Robotics 2020. Multi-Annual Roadmap.* Release B 02/12/2016. Author.

Taggart, W., Turkle, S., & Kidd, C. (2005). An interactive robot in a nursing home: Preliminary remarks. Workshop: Towards Social Mechanisms of Android Science, Stresa, Italy.

Velasco-Alvarez, F., Ron-Angevin, R., & Lopez-Gordo, M. A. (2013). *BCI (Brain Controller Interface) -based Navigation in Virtual and Real Environments* (Vol. 7903). Advances in Computational Intelligence. Springer Lecture Notes in Computer Science.

Zielinska, T. (2006). Control and navigation aspects of a group of walking robots. *Robotica*, *24*(1), 23–29. doi:10.1017/S0263574705001840

Zielinski, C. (2010). Robotics - Quo vadis? *Journal of Measurements and Control,* *5*, 9-19. (in Polish)

Zielinski, C., Stefańczyk, M., Kornuta, T., Figat, M., Dudek, W., Szynkiewicz, W., ... Iturburu, M. (2017). Variable structure robot control systems: The RAPP approach. *Robotics and Autonomous Systems*, *94*, 226–244. doi:10.1016/j.robot.2017.05.002

Chapter 7
Overview of Wireless Sensor Network, Robotics, IoT, and Social Media in Search and Rescue Activities

Sarah Allali
University of Science and Technology Houari Boumediene, Algeria

Mahfoud Benchaïba
University of Science and Technology Houari Boumediene, Algeria

ABSTRACT

In recent years, many researchers have shown interest in developing search and rescue systems composed of one or multiple robots. To enhance the robotic systems, wireless sensor networks and internet of things (IoT) were integrated to give more awareness of the environments. Additionally, data exchanged in social media during emergency situations can help rescuers, decision makers, and the public to gain insight into the situation as it unfolds. In the first part of this chapter, the authors present a review of robotic system and their environments in search and rescue systems. Additionally, they explain the challenges related to these systems and the tasks that a robot or a multi-robot system should execute to fulfil the search and rescue activities. As a second part, the authors expose the systems that integrates WSNs and IoT with robots and the advantages that brings those. Furthermore, they expose and discuss the remarkable research, the challenges, and the open research challenges that include this cooperation.

DOI: 10.4018/978-1-5225-5276-5.ch007

1. INTRODUCTION

Search is the activity of finding survivors, victims or a searched object; while rescue is the activity of extricating survivors, victims or a searched object. The search and rescue activities start when the rescue center sends a team of rescuers to inspect the disaster area. In this step, the team search for victims via a systematic or a random search and collects other information about the disaster area that may affect the rescue operations for example: obstacles, falling debris, toxic gases…etc. After this step, the rescue center plans the rescue activity by creating a map of the disaster area that contains the location of victims. This helps them to excavate rubbles and rescue victims. Finally, victims are transferred to hospitals and the disaster area is put under surveillance for any changes.

To execute these tasks, researchers proposed to add robot to help in the search and rescue activities. Then, to achieve this mission quickly, efficiently and accurately a cooperation between robots was proposed. This way of search and rescue activities will fulfil in a team mission. This team cooperation could replace the need for human presence in the disaster site for almost all the steps of search and rescue operation cited previously. However, there are some challenges that this system faces, these challenges are related to the mobility, communication, control, robot's sensors, power and the human robot interaction (HRI). Also, there are some challenges related to the reliability of the system such as: optimizing mission times, robot localization, creating the disaster area map, decision making, improving robots reliability, finding the maximum number of victims, sending the maximum of information to the rescuers (see Murphy (2008)).

To overcome some of these challenges, an inclusion of stationary sensors in the system was proposed; those sensors help to coordinate between robots and make them more aware of the surrounding environment. These sensors do several tasks to help robots in their missions. They could be relays that send the information collected from robots to the operator. They could be also coordinators that transmit the data between robots to extend robots coverage. They could help to have a distributed tasks allocation (For example, one sensor choose the robot that move to a destination). The use of sensor network could help to solve the problem of resource conflicts by routing data in different paths. Additionally, having a wireless sensor network (WSN) is the best way to monitor the whole disaster area and to detect the different events that might happen at any time. Furthermore, the WSN track victims and robots to communicate their location in real time to the rescuers. However, this wireless sensor network has several challenges related to the hardware of sensors used, to the energy consumption, to routing protocols, to the degree of autonomy and to the quality of services.

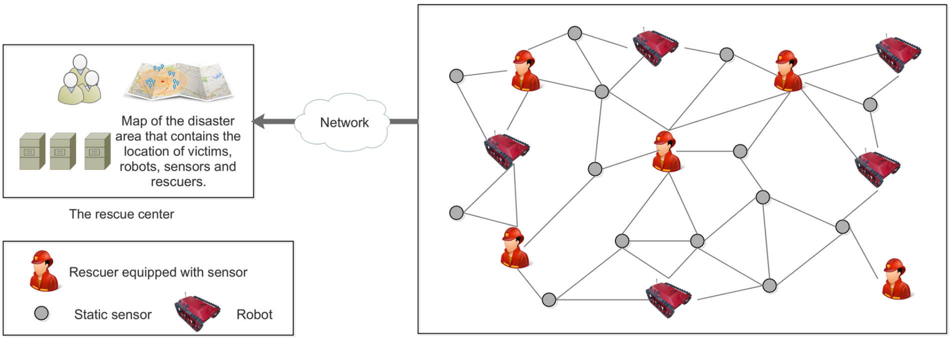
The architecture of a search and rescue system as shown in figure 1 assumes that the architecture is divided into two parts: the rescue center and the disaster area network. These are related via a wired or wireless network dependent to the architecture of the application. Rescuers provided with powerful servers supervise the rescue center. The second part of the system, which is the disaster network, was composed in the beginning by single powerful robot only. After that, the rescue application moved from the use of one robot to the use of multi-robots. After that, an inclusion of a stationary sensor network was necessary. This latter helps the mobile robots to communicate between them and obtain additional sensory environmental information. These sensors are assumed to cover the whole area during all the search and rescue activity. For this, the sensors could be deployed before or after the disaster. The deployment after the disaster could be done by using a robot that distribute sensors through the area randomly. The deployment before the disaster could be done by integrating sensors in buildings structure.

In this article, we present a review of robotic systems in search and rescue field area and we explain challenges related to these systems and tasks that a robot or a multi-robot system should execute to fulfil the search and rescue activities. Then, we expose a system that integrates WSN with robotic system in order to better accomplish rescue tasks.

Furthermore, we expose some researches that include this cooperation and the remaining open issues.

The article is organized as it follows: In section 2, we discuss the robotic system related to the rescue. Section 3 defines the network robot system and discusses the different tasks that a multi-robot system or a robot accomplish during the search and rescue mission. In section 4, we present the different WSNs applications that are used during the search and rescue activities. In section 5, focus our interest on the use of a collaboration of a wireless sensor network and robots to achieve the search

Figure 1. Architecture of a search and rescue system


Map of the disaster area that contains the location of victims, robots, sensors and rescuers.
Network
The rescue center
Rescuer equipped with sensor
Static sensor
Robot

and rescue task. In section 6 we present the advantages of using IoT and social media in search and rescue application. Additionally, we present the different existing works in the literature. The last section is a discussion that exposes challenges and the remaining open issues that appear due to the use of a wireless sensor network.

2. ROBOTIC SYSTEM

In this section, we discuss the robotic system related to the rescue. For this, we define tasks that have to be done by the robot, the different types of robots used in search and rescue activity, the different sensors equipping robot. Additionally, we expose challenges related to the communication, survivability, victims' localization and multi-robot systems. Finally, we discuss the multi-robot tasks and motion planning.

2.1. Rescue Robots Tasks

Types of tasks that have been proposed for rescue robots are described below:

- **Search:** In this task, the robot goes on a blind search for victims and hazard accurately, rapidly and without creating risks on victims and rescuers.
- **Exploration and Mapping:** In this task, the rescue robot explores randomly the disaster area and creates a map of it. Additionally, the robot finds and locates victims in the created map.
- **Path Planning:** In this task, the robot tries to understand the potential danger in each area of the disaster place to give the best path to rescue victims and also to move robots.
- **Relay or Coordinator:** In this task, the robot has the mission of linking robots to increase the robot coverage area and to coordinate tasks between the different rescue robots.
- **Medical Support:** In this task, robots try to send to the rescue center the medical condition of victims and provide a medical support while victims are in still trapped in the disaster area.
- **Rescue and Evacuation:** In this task, the robot guides victims to the exit through the safest paths.

2.2. Robot Sensors

To achieve the search and rescue task, each robot is equipped with a set of sensors. These sensors are used for the control of the robot (robot status, speed, position… etc.), the inspection of the environment (air quality, temperature, humidity, radioactive substance) and the detection and tracking of victims (Cameras, laser rangefinders, ultrasonic range finder, global position system (GPS) sensors, contact sensors, force sensors and infrared sensors). An interesting survey of methods for detecting human presence was proposed in Teixeira et al. (2010). This article classified the victim detection methods into based in: binary sensors, vibration sensors, Radio, Ultrasound, Laser and cameras.

2.3. Survivability and Mobility

The rescue robot is supposed to survive to dangers of the search and rescue environment that may have toxic liquid, dangerous gas, radioactive substances, biochemical materials, extreme temperature and falling debris. Additionally, the robot should have a flexible software and hardware to be able to take decisions in different cases, operations and regions (see Liu et al. (2007)). In the software flexibility, the robot should have enough intelligence to respond to challenges and instable factors in the environment with adding some autonomy to robots.

The hardware flexibility depends on the mobility of the robot. The inspected space is complex because it represents an unpredictable combination of vertical and horizontal elements with unknown surface characteristics, obstacles and inaccessible areas (see Murphy (2008)), additionally the disaster ground is usually soft for dusts, muddy for water and rugged for debris. Thus, the robot should overcome these obstacles. The physical design of the robot is important for its mobility. The robot should be small enough to easily move, but it should be large enough for equilibrium to avoid to be reversed (see Liu et al. (2007)). In addition, the stability and the self-adjustability are very important to avoid dropping the robot. The robot should be fault-tolerant and fault-handling, in case the robot or a part of the robot get damaged a solution for this problem is to have a reconfigurable modular search and rescue robot.

2.4. Multi-Robots

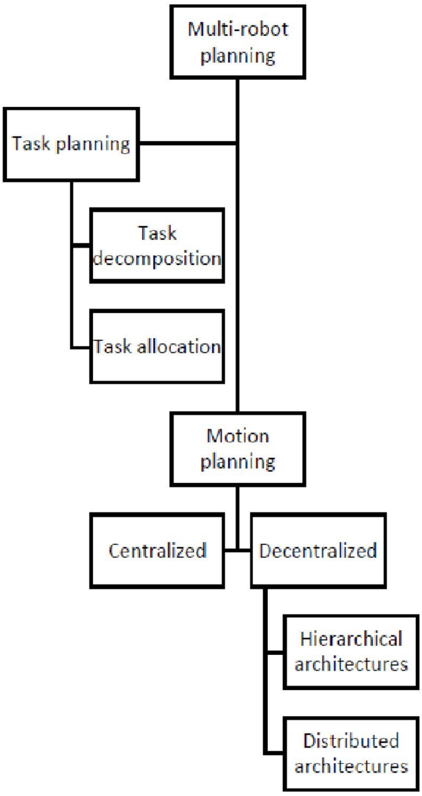
The use of multi-robots makes possible the use of data fusion, information sharing and fault-tolerance that introduce robustness to the system. Some complex tasks cannot be achieved by one robot, but need the use of a multi-robots system that combines several robots with diverse abilities. Even, for applications that can be achieved by the use of only one robot, the use of multi robots gives a reliability, flexibility, scalability and versatility to the system which concludes to a better overall system performance (total time required, less energy consumption…etc.). Additionally, using a group of simple, cheap robots may be lower cost than using one powerful, complex and expensive robot.

2.5. Multi-Robots Task and Motion Planning

Multi-robot planning is the task of coming up with sequence of actions that will achieve a goal. Multi-robot planning as shows Figure 2 is divided into task planning and motion planning (see Yan et al. (2013)). Task planning solves the problem of which robot executes which task, this includes: task decomposition and task allocation. Task decomposition is the phase that defines the decomposition of the mission into several single subtasks realized by a single robot to achieve the global mission (see Chen et al. (2010)).

Task allocation is an instance of the optimal assignment problem in which there is a number of agents and a number of tasks. Any agent can be assigned to perform any task, incurring some cost that may vary depending on the agent-task assignment. It is required to perform all tasks by assigning exactly one agent to each task in such a way that the total cost of the assignment is minimized.

Figure 2. Classification of multi-robot planning



Motion planning generates the path of each robot by taking into consideration the paths of others in order to avoid any collision, congestion or deadlock.

The decision making guided by planning can be centralized or decentralized in accordance with the architecture of robots.

Decentralized architectures can be divided into two categories: distributed architectures and hierarchical architectures.

The decentralized architecture can better respond to unknown or changing environments, and usually has better reliability, flexibility, adaptability and robustness.

3. ROBOT AND MULTI-ROBOT SYSTEMS

An important application of network robot system is the search and rescue activities. In this section, we discuss the different tasks that multi-robots system or a robot should do to accomplish efficiently the search and rescue mission.

3.1. Exploration

Within the search and rescue application, the first task that should be realized is the exploration of the unknown environment. One robot or a cooperative group of robots could achieve this. The goal of any exploration strategy is to minimize the overall exploration time. Thus, it is essential that robots keep track of which areas of the environment have already been explored. Additionally, robots have to construct a global map in order to plan their paths and to coordinate their actions.

1. Exploration with one robot:
    a. **Simple Maze Exploring:** Maze exploring algorithms try to achieve two objectives: search for an object hidden inside the maze or try to escape out of the maze from an unknown location.
    b. **Using Touch Sensing:** Is a non-heuristic algorithm for exploration in unknown terrains.
    c. **Navigation Using Vision:** In this navigation, the robot uses a camera or a laser to inspect the environment.

The algorithm of each strategy cited above guarantees only to solve the exploration problem. Based on these works, some advanced works try to guarantee a degree of performance, lower bond on the distance traveled, optimal scan operation and path planning. In addition, these algorithms do not take into consideration the resource conflict.

2. **Exploration via Multiple Robots:** The key problem to be solved in the context of multiple robots is to choose appropriate target points for each robot. This let them explore different regions of the environment simultaneously with avoiding: collision, overlap, congestion and deadlock. The main existing cooperative exploration strategies are:
    a. **Frontier-Based Exploration:** The idea is to gain the most new information about the explored space, by moving to the frontier and diffuse the information to other robots.
    b. **Marker-Driven Exploration:** This strategy is based on dividing a mission into sub-missions and robots place markers on these sub-missions.
    c. **Role-Based Exploration:** Robots in role-based exploration have two roles: explorers and relays. The role of explorers is to explore an unknown region by using frontier-based exploration strategy. The role of relays is to maintain connection between the base station and explorers.

3.2. Simultaneous Localization and Mapping (SLAM)

One of the important robot mission is to generate maps of its environments and localize its position as well as victims and objects of interest positions (Nagatani et al. (2013)). Human rescue workers for victim retrieval efforts can use these Maps. SLAM is the process that allows a mobile robot to build from an unknown environment a map and uses it to track its location in real-time. Many techniques were proposed to solve the SLAM problem, the principal solutions are: EKF-SLAM, FastSLAM, L-SLAM and GraphSLAM. Since 2005, researchers interested widely to VSLAM (visual SLAM) using primarily visual (camera) sensors. According to Milford and George (2013), there are many vision-based techniques including FAB-MAP, MonoSLAM, FrameSLAM, V-GPS and Mini-SLAM that are competitive with or superior to range sensor-based algorithms.

3.3. Robot Control

The existing search and rescue approaches was developed with low-level robot control, semiautonomous control and multi-robots control in order to improve the performance of rescue robots and minimize the workload of robot operators. This is classified into categories:

1. **Non-Autonomous System:** The non-autonomous robot means to have a fully teleported system, which needs to be guided by the operator to explore the

supervision area, and thus requests many messages exchange, which exhausts the battery of the robot quickly.

2. **Low-Level Robot Autonomy:** Some operations were added to the robot that make a low-level of robot autonomy. That extend the life service of the robot by reducing the data exchanged and optimizing the time wasted in communication. For example: self-traversing Stairs and Uneven Terrain, and self-correcting in uneven terrain.
3. **Semi-Autonomous Control:** Semi-autonomous system mixes human tele-operation with robot autonomy, which contains two categories: fixed and adjustable autonomy. In robotic semi-autonomous control schemes with fixed autonomy, a robot focuses on low-level tasks, and the human operator is in charge of high-level control and supervisory tasks (see Okada (2011)). In adjustable autonomy, the robot could move to a fully tele-operated mode or a fully autonomic mode depending on the operator request and according to the situation.

3.4. Search and Rescue

Multiple-mobile robots play an important role in disaster recovery and was widely used in this filed, in the following we will expose some existing search and rescue multirobots system. A search and rescue system was proposed in Loukas et al. (2008). This work, deals with the fundamental problem of the optimal allocation of search and rescue robots that need to form an ad hoc network with all or a group of victims and connect them via a multi-hop to a wireless sink. To solve this problem, authors proposed to adopt the solution for the problem of maximizing area coverage while maintaining connectivity, where they deal with the connection of civilians instead of area coverage. The work in Timotheou and Lokas (2009) presents a distributed algorithm, which involves clustering possible locations of civilians according to their expected shortfall, which facilitates the connectivity between victims and the exploration activity. The algorithm bases on minimum spanning tree, which gives an efficient allocation in terms of time and energy. Comparing to solution proposed previously, Asakura et al. (2010) proposed an ad-hoc unicursal protocol for communication systems in disaster situations. This protocol is designed for gathering information about rescues effectively in a big earthquake. In this protocol, terminals are connected to each others in a linear form, which makes the communication loads among terminals relatively equal. Additionally, they introduce virtual terminals, in which two neighboring terminals activate alternatively which decreases the frequency of packets.

4. WSNS FOR EMERGENCY RESPONSE, SEARCH AND RESCUE ACTIVITIES

WSNs have been widely used in emergency response systems due to their reliable infrastructures that could handle the frequent and unexpected changes in the deployment environment. In fact, WSNs can report in real-time about every change in the environment, from a simple hazard detection to a human victim detection, which helps to fulfill the emergency response tasks. In this section, we present the different WSNs applications that are used during the search and rescue activities.

4.1. Monitoring

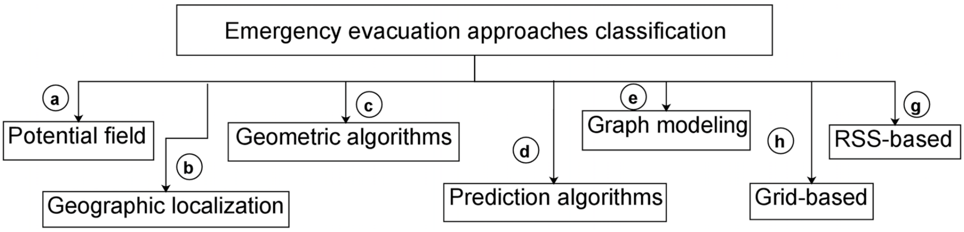
An important task in the search and rescue activities is related to supervising the evolution of an area of interest, which is done thanks to the monitoring system. Monitoring is the act of collecting information concerning the characteristics and status such as: evolution of danger, victim detection, and robot or victim movement tracking.

4.2. Emergency Evacuation

The emergency evacuation activity in WSNs tackles the problem of evacuating victims to an exit through the shortest and safest path. The main problem to solve is to find a route to move inside the area of interest while avoiding obstacles and hazard locations. We propose to classify emergency evacuation system as follow (See Figure 3):

1. **Potential Field Approach:** This approach considers sensors as containers of a potential field points, the target as a point in the potential field that combines
2. **Geographic Localization Approach:** In this approach the sensors need to know their location e.g. by using GPS.

Figure 3. Emergency evacuation system classification



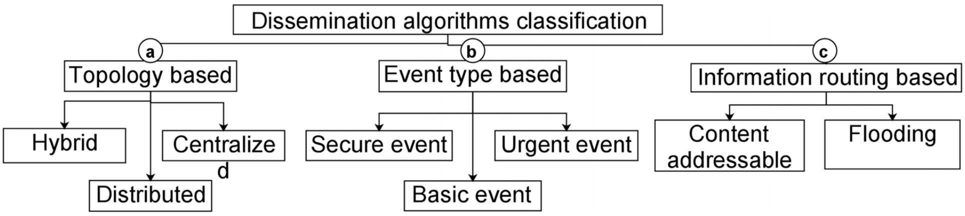
3. **Geometric Algorithms Based Approach:** This uses properties of geometric graph to plan evacuation path as far from hazards as possible.
4. **Prediction Algorithms Based Approach:** This calculated the navigation path based on the longest escape time before the hazard reaches the navigated object.
5. **Graph Modelling Approach:** In this approach the WSNs are represented by a graph where nodes on the graph represent sensors and an arc between two nodes represent a physical path between these two nodes. Arcs could hold weight on it, this could be (1)the time to move from one node to another, (2) the time for the hazard to emerge in the path that relates this two nodes and/ or (3) the risk rate in this path.
6. **Received Signal Strength (RSS) Based Approach:** This uses RSS to know the location of target and to navigate them to a goal.
7. **Grid-Based Approach:** This approach overlies a grid on the monitored area; this grid is composed of cells where each cell contains at least a sensor. The victim or the robot is allowed to move to adjacent cells if there is no obstacles between them. Search algorithms are used to find a path from the location of the target to its goal.

4.3. Event Detection and Dissemination

Other important tasks in the search and rescue activities are related to the detection of events and the dissemination of information about the existence of those events. A good WSN for search and rescue activities should accurately, effectively and promptly detect and transmit information about the existence of events (hazard, victim..etc.). Different information dissemination algorithms have been proposed. This could be classified into three categories: topology based, event type based and information routing based (See Figure 4).

1. **Topology Based:** The topology of the network affect the way the data is disseminated through it. Depending on the architecture of the WSN (centralized, distributed or hybrid) the data dissemination method changes. The centralized network is composed of sensors and sinks that transmit the information to the rescue center. In this architecture, sensors send information to the sink that send it to the rescue center. Distributed architecture has only sensors, which have the same role. This architecture uses usually dissemination through the whole network or through area of interest to send a certain information. The hybrid architecture, is a mixture of the two previous architectures which make this later use dissemination methods that are used in both of them.

Figure 4. Dissemination algorithms classification



2. **Event Type Based:** The information of an event is disseminated differently according to the type of the event. The classification of dissemination algorithms based on the type of events may be classified on three categories: Secure event notification, urgent event notification and basic event notification. In the secure event notifications, data are routed through specific routes only, which have defined encryption rules. In the urgent event notification, data are routed through routes that guarantees a quick and safe data transmission. In the basic event notification, data are transmitted through the route that is indicated by the routing protocol.
3. **Information Routing Based:** The routing protocol used in the WSN, decides in which way the data is disseminated through the network. Routing protocols that are used for data dissemination could be classified into: flooding based and content addressable based. In the flooding based the data are transmitted through the whole network. In the content addressable based the data is transmitted to a specific destination using routing tables.

5. ROBOTS AND SENSORS

The use of a team collaboration between robots and sensors in the search and rescue activities attracts researchers' interest recently. Some works have been proposed recently around this idea. In this section, we present the different advantages those sensors networks could bring to the robot and multi-robots system in a search and rescue system. This section exposes as well the some existing works in the field, challenges and the remain open issues.

5.1. Hybrid Sensor Network

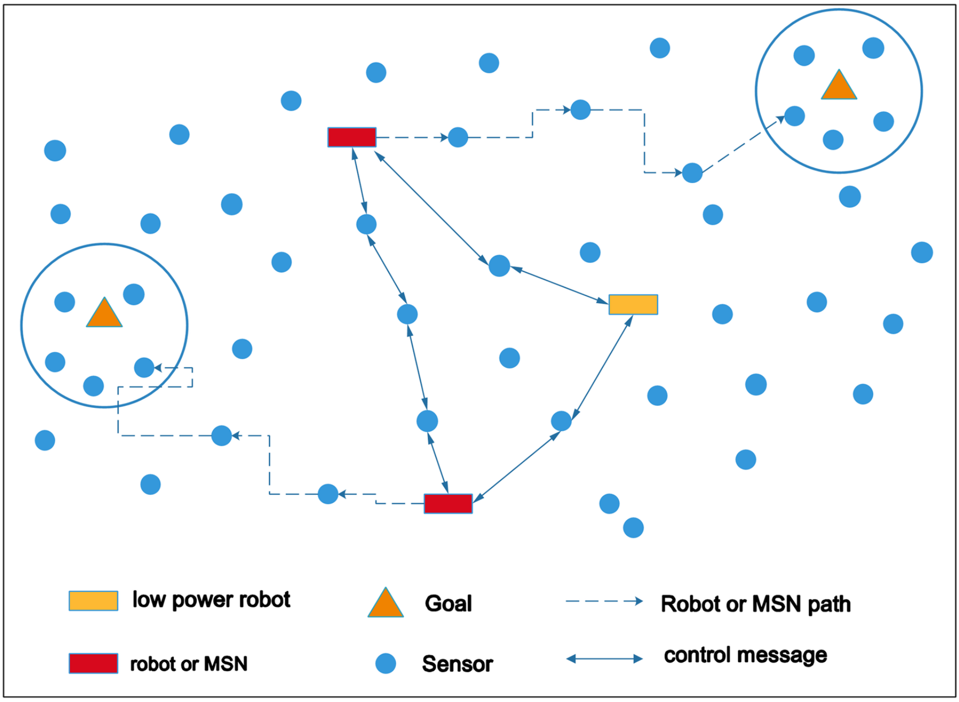
A hybrid sensor network is composed by static sensor nodes and mobile sensor nodes or robots, which cooperatively perform tasks like environment monitoring, collaborative sensing, target tracking…etc. Figure 5 represents a typical hybrid sensor network with communication between three robots for the moving robot selection and a movement of two robots to their goals.

In this figure, the small circles represent static sensors, these are equipped with different types of sensors, such as, temperature or humidity sensors, fire detecting sensors, seismic sensors, motion detectors,…etc.

The rectangles represents mobile sensors or robots that move to a goal. These nodes are capable to move, have more energy and longer communication range than static sensors, are equipped with obstacle detection sensors, can be equipped with sensors to detect their location such as GPS or sonar receivers, also can be equipped with cameras.

The triangles represent an event or a goal that should be reached. The group of sensors that sense and surround the goal are represented with a circle that is surrounding the goal. Those sensors have the task of communicating an alert signal through sensors network. As we see in the Figure 5, robots exchange messages via the sensor network that coordinate between them and select the robot that should move to each goal.

Figure 5. A hybrid sensor Network



5.2. Tasks

During the search and rescue mission, many tasks should be achieved. Tasks allocation to robots could be done in a distributed way between sensors or in collaboration with robots. Task allocated to sensors in search and rescue system are explained in details in the following points:

5.2.1. Relay

The sensor networks could be the relay that: send the information collected from the robot to the operator and transmit messages between robots which will solve the problem of communication between robots. In this case, the robot is equipped with two interfaces, one to communicate with the robot and the other to communicate with the wireless sensor network (IEEE 802.15.4 or IEEE 802.11b/g/n).

Normally, the WSNs were designed for non-frequent data transmission applications such as events detection or periodic measurements. The application used for the search and rescue requires frequent data transmission, which consumes energy very quickly. To deal with this problem, authors of Tuna et al. (2014) proposed to investigate the relationship between data transmissions and the lifetimes of different types of sensor nodes. Indeed, it is not only the material used in the application that matters but also the routing protocol, the logical architecture of the network and the use of the aggregation of data. All these provide a load balancing between sensors of the network and therefore a longer lifetime of sensor network with respect to the connectivity and the coverage of the network.

5.2.2. Coordination

An important task that could be done by sensors is to coordinate with robots. Shan and Tan (2006) cited the following sensor and robot assignments during coordination:

- Sensor nodes can be used to extend the robot system’s coverage.
- Sensor nodes and robots can construct a map to rescue human victims.
- Sensor nodes are used for maintaining environment map and communication, while robots as explorers and guiders for human victims.
- Sensor networks can be used for robot navigation across an unknown environment.
- Sensor networks can also intelligently select some mobile sensors to improve the system’s sensing capability.

5.2.3. Fixing Robots Resource Conflicts

Resource conflict appears when many robots try to access to the same resource at the same time. The resource could be a space, a manipulated object or a communication media. The problem of communication media sharing is associated with bandwidth limitation (see Yan (2013)). Some solutions were proposed to solve this problem such as in (Nerurkar et al. (2011)). Where authors proposed the use of sensor network to help to solve this problem by routing data in a different path.

5.2.4. Task Allocation

A task could be realized by one robot or by a group of robots or in coordination between sensors and robots. For this, a task could be divided between robots and sensors into independent subtasks, hierarchical task trees, or roles. Subtasks and roles could be done independently but hierarchical task trees should follow the tree hierarchy i.e. the task could be done only when the previous task in the line of the hierarchy is completed. Tasks should be shared in way to minimize the global cost (i.e.: minimize the movement of robots and minimize the communication between robots and sensors), this problem is strongly NP-hard. The difficulty of task allocation problem appears in the: selection of the number of robots or sensors required per task, number of tasks that could be done by a robot or a sensor at the same time, the coordination dependencies between tasks, allocation of tasks to frame of time during mission time.

5.2.5. Event Detection

Sensors are equipped with devices that allow them to sense and communicate with each other which lead to the detection of events and unusual data behaviors in a monitored environment (see Nasridinov et al. (2014)). To detect effectively the events, the sensor should have a high true detection rate and a low false alarm rate and it should take advantage of timeliness. Event detection presents several challenges. In Kerman et al. (2009), authors classified this principal challenges: situational dependence, criticality of application, numerous and diverse data sources and network topology.

Additionally, Nasridinov et al. (2014) classified event detection methods into three categories: statistical, probabilistic, artificial intelligence and machine learning.

5.2.6. Target Tracking

Target tracking is an important application of Wireless Sensor Network and Robots (WSNR), which is set up in areas of surveillance field, survivor detection, localization and robot tracking. Various approaches have been investigated for tracking targets, considering diverse metrics like scalability, overheads, energy consumption and target tracking accuracy.

5.2.7. Intelligent Monitoring System (IMS)

WSNs could form a good IMS that, is used for various monitored environments and monitoring missions such as coal mine monitoring, underground structure monitoring, and fire detection and rescue activity. For instance, for the fire detection, the system uses sensors to detect abnormal temperature and smoke intensity. This, automatically triggers the fire alarm and notifies fire fighters. Additionally, sensors report to the fire fighters in real-time the environmental changes and activities of humans and generate a real-time fire map which reflects the situation of the building in real time. This Map is used also to find trapped victims and the most efficient route to evacuate victims. In addition, An IMS can be used in various fields to reduce and prevent loss of life and goods when dangerous events happen.

5.3. Missions

In this section, we discuss the different missions that are fulfilled by a cooperation of a wireless sensor network and robots. We discuss some existing works in each mission and the possible open issues.

5.3.1. Emergency Evacuation

A WSN is widely used in emergency evacuation missions.

In Gelenbe and Wu (2012), authors proposed a scheme of sensor-aided Cyber-technical systems that enables fast and intelligent response to emergency situations which require quick response and real-time monitoring. In this article, authors discuss three classes of distributed protocols to compute safest evacuation paths.

The first class of approaches is the potential-maintenance approaches, these approaches adopt in a distributed manner the concept of artificial potential fields which is used in mission planning. In this, exit sensor generates an attractive potential and obstacles generate a repulsive potential. In this way, sensors are pulled to the exit and pushed away from obstacles. Based on this information, each sensor computes an overall potential value, which evacuates victims.

The second class is the geometric approaches; in this class, approaches use geometric graphs to plan evacuation paths.

The third class of approaches is the prediction-based approaches. This kind of approaches predict how long a hazard will take to reach the sensor. This helps to compute the evacuation path with the longest escape time before the hazard reaches the sensor.

In Liu and al. (2016), a framework for calculating efficient routes to evacuate survivors during a fire was proposed. The framework is composed of: a sensor network, a communication network and a server. The sensor network comprises a group of sensors that divides the monitored area into sub-areas where a sensor should sense whether this area is blocked or transitable. These information are communicated to the server through the communication network. The server aggregates the data collected and calculates the evacuation routes by using A* algorithm.

We noticed two limitations related to this solution, the first limitation is related to the use of A* algorithm. Since A* uses an exponential space, the processing time to find the solution maybe exponential too. We know that processing time is critical in search and rescue application, if it is long this may lead to the death of the occupant. The second limitation is related to the scenario studied in this solution, the solution does not take into consideration the scenario of more than one victim that tries to escape in the same time. In this case, we could have a congestion of citizen.

5.3.2. Coordination and Navigation

Coordination and navigation show the importance of sensors in search and rescue activity. This gives a platform to easily coordinate between mobile robots and also to easily navigate them in the disaster place.

In Jiang et al. (2011), authors propose a suite of three schemes for mobile robots coordination and navigation in positionless wireless sensor networks with the help of directional antennas. The first scheme is the furthest node forwarding (FNF). In this scheme, node that detect the victim or an event is called: waiting-for-rescue (WFR) node. When this node detects the event, it broadcasts event notification packets to the entire network. The nodes that participate in the forwarding activity represent a tree, rooted at WFR. This structure is also is used in the two other schemes.

The second scheme is mobile robots coordination (MRC), this is designed to coordinate multiple WFR nodes and multiple mobile robots. The scheme is distributed and greedy based.

The third scheme is tree-assisted navigation (TAN), in this scheme the nodes guide the mobile robot to reach the WFR node using a directional antenna without position information. There is a unique path which take the mobile robot to the WFR node.

The simulation results shows that FNF scheme achieves high reachability with very few rebroadcast packets compared with other schemes. The simulation results shows that mobile robots can be successfully navigated to WFR nodes for all simulated scenarios.

The proposed solution is interesting; however it has some drawbacks. The first is related to the FNF that does not study the case when two sensors detect the same victim or events in this case we will have in the network two different alerts for the same victim. In addition, the FNF focuses only on choosing a set of nodes with the smallest distance to the robot while they do not take into consideration the energy of the selected node. Moreover, the process selects only one path and uses it for all the process while this path might get damaged in the middle of the process.

Authors take into consideration only the distance factor and ignore an important factor, which is the time for example time to chose the robot that participate in the navigation.

Allali et al. (2017) proposed an overall system that aims to help robot to rescue immobilized victims through the safest and shortest paths. This is done by using a distributed decision making system that considers hazard location in the area. The system is composed of several robots that have the mission of rescuing victims. Sensors have three missions: (1) detecting the existence of victims or any hazard in the area, (2) disseminating information about hazards and victim, (3) guiding robots / rescue agents to their destination (victims to the exit). The proposed process takes into consideration the priority between victims by making the robot / agent choose between different calls for rescue. Also, it gives the target the right to choose the most appropriate robot / agent to answer the call of rescue. The communication between robot and sensors makes only the most suitable free robot / agent move to the nearest and urgent target.

Allali et al. (2018) proposed a No-Collision Grid Based broadcast scheme (NCGB) that disseminate information about victims while creating all possible paths from the victim to any cell in the grid. This helps in guiding the robot through the safest and shortest route. In the navigation system proposed the robot schedules the sequence of victims to rescue and then moves to the list of victim. The proposed system aim to increase the number of victim saved. For this, authors proposed an Ant Colony System with Victim Lifetime Window (ACS-VLW) to solve the problem of navigating robot. The required navigation will browse immobilized victims using the shortest and safest path. This ensures an increase of the number of rescued victims.

5.3.3. Victim and Event Detection

An important task that could be achieved by a sensor network more efficiently than a multi-robots system is victim and event detection.

Yuan et al. (2011) proposed one of the largest sensor and robots network for search and rescue. Authors of this article have designed and implemented smart mobile robots and a static deployment of sensors that cover the entire area with a strong network connectivity. Authors proposed an event detection protocol that saves energy with an accuracy rate approaching 100%, thanks to the protocol wake-on sensor TELOS W. In addition, a navigation protocol is applied to guide the robot (TelosW - Bot) and therefore guides victims to the nearest exit, through the most secure way. The navigation protocol is based on a Free-GPS navigation strategy with a distributed decision-making.

This work has two main limitations, the first is that it covered only the energy consumption related to the event detection and neglected the energy consumption caused by the messages exchanged. These messages are generated by the distributed decision making navigation protocol and the event detection process. To reduce this exchange of messages, it could be possible to aggregate the data collected by the event detection before sending it.

The second limitation of this work is related to the assumption of prior knowledge of the deployment map. Moreover, they suppose that the proposed deployment architecture allows full coverage of the field of supervision while it is not realistic.

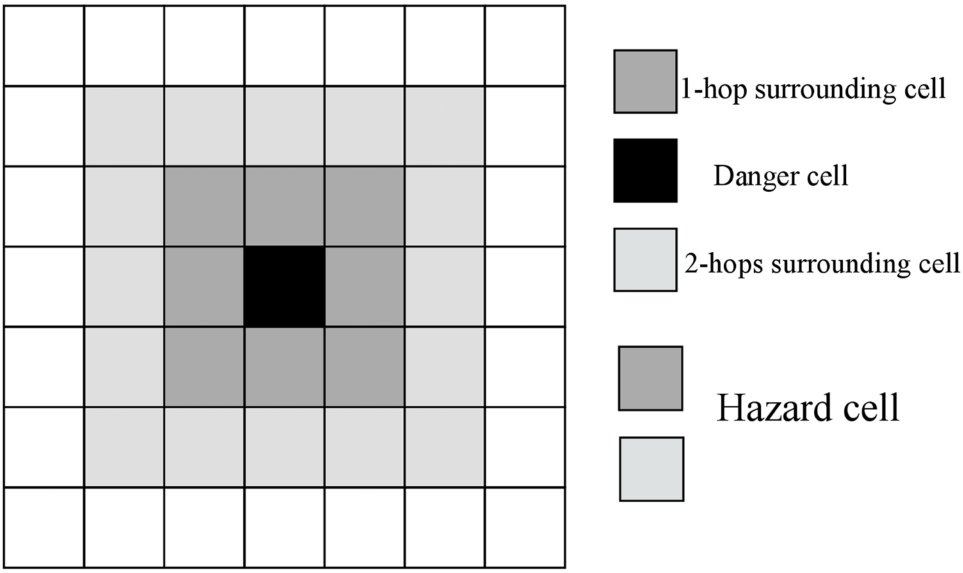
Allali et al. (2016) proposed a grid-based architecture to monitor an area of interest that could be of a large size through number of sensors deployed randomly to cover the whole area. In this solution, the monitored area is represented by a grid that is logically divided into cells and in each cell at least one sensor is deployed as shown in the figure 6. The status of sensor located within a certain cell tells the situation of its cell and other cells around. Sensors could be in hazard, safe or danger status depending on its situation. A sensor, normally, is in the safe state and it changes its state to danger if there is any danger accruing in its cell. In addition, all cells that surround this cell with one to two hops are defined as Hazard cells as represented in figure 6. For this, deployed sensors enter into a dynamic cooperation to ensure an energy-efficient dissemination of the whole area with an alert about the new hazard. The proposed solution has been validated through

intensive simulations, which show an important improvement in terms of overall network overhead as well as overall network energy consumption

5.3.4. Tracking

To fulfil the search and rescue task, the rescue system must be equipped with a tracking system. This latest should have the ability to track the positions of moving victims and to trace robots' positions even when they navigate underneath the rubble. Since the GPS system does not work correctly under rubbles and the communication with the rescuer is limited, tracking network is required. This network is based on

Figure 6. Grid representation and neighborhood of a danger cell



a group of wireless sensors connected to each others and to the rescue central to carry out the tracking task. This tracking network could be stationary or mobile.

Authors of Ko et al. (2009) proposed a design and an application of a General Suppression Control Framework (GSCF) based on distributed wireless sensor network prototype system for tracking mobile search and rescue robots. The purpose of this physical prototype system is to demonstrate the possibility of applying advanced ZigBee sensors to form a network that can locate a small group of mobile robots within the wireless sensor network. Additionally, this system tests the mathematical model of the proposed architecture.

5.3.5. Isolated Safe Areas Detection

In Ai et al. (2017), authors address the detection of isolated safe areas issue. Authors defined an isolated safe area as a safe area, which is contained by a dangerous area. Detecting this safe area is an important task since it could contain trapped victims; also, victims could be guided to this safe area until the arrival of rescuers. However, this areas are temporary safe because dangerous areas may spread quickly for example fire propagation. To detect this isolated safe areas, two algorithms were proposed: centralized and distributed.

In the centralized method, the base station of the wireless sensor network sends queries to find the grids that the event is happening in and the situation of this latest.

The advantage of the centralized detection is the easy implementation and its time complexity; however, this algorithm wastes time while it is waiting for sensors to finish processing the event queries.

The distributed process allows discovering isolated safe area as early as possible since sensors do it in a distributed way. Also, when sensors fix the boundaries of one of the isolated safe areas, they send it directly to the rescuers. In this situation, the area of interest is divided into subareas, which are represented by a reporting tree. In this reporting tree, the internal nodes receive reports from their children and merge the received result with their current situation to have a partial result, this partial result is sent then to their parents.

This work has some drawbacks, first the size of the sub-areas was not fixed, a question that is still open is what is the best size of the sub-area to do the task in the shortest time. In addition, another open issue is to compare the distributed algorithms, used in this article to other distributed algorithms since authors did not justify their choice nor do the comparison. Another proposition to collect information about the safe areas is to formulate clusters and choose a cluster head to communicate the information to the rescuers. This solution might collect information about safe areas more rapidly than in the case of the tree based and might waste less energy.

Another open issue in this work is to adopt this work to a natural disaster such as earthquakes and tsunamis. In this kind of situation, safe isolated area is used as safe areas where victims should be guided in.

6. INTERNET OF THINGS AND SOCIAL MEDIA FOR SEARCH AND RESCUE APPLICATIONS

In this section, we present the advantages of using IoT and social media in search and rescue application. Additionally, we present the different existing works in the literature.

6.1. Internet of Things (IoT)

IoT devices operating in emergency situations are used to help both WSNs and robots to accomplish the search and rescue tasks. IoT devices helps WSNs to collect emergency information and thanks to the internet, IoT distributes the captured data to the required sources promptly. Since the default communication infrastructure may fall in disasters, IoT is an ideal alternative to relay critical information. Additionally, the use of IoT in rescue application may help robots. This could help to coordinate between robots and the rescue center, it could help also to supervise and monitor the navigation of robots.

Chen et al.(2011) integrated IoT technologies in earthquake rescue. Authors proposed a method that predicts the nature of an earthquake and its economical losses. The proposed method is used during the post-earthquake activity where the data coming for the IoT are exploited to supervise the earthquake evolution. Moreover, the data is exploited to perform decision-making activities.

Li et al. (2014) proposed an IoT-based framework that collect real-time data and make scientific decisions based on the sufficient and reliable data in emergency response systems. Additionally, they developed a consensus model and a procedure to achieve the consensus group decision promptly.

Wang and al. (2017) proposed to improve the performance of firefighters by using IoT. The proposed solution used the quantitative method to gain insights into the usefulness of IoT as an aid to firefighting. A Monte Carlo simulation was developed for processing the detailed firefighting interactions in situations of uncertainty. The result of simulation showed a significant improvement when the IoT was implemented in firefighting.

6.2. Social Media

Social media represent a rich source of real-time information about real-world events. This could be a good source of information collection and diffusion during mass emergencies. The information diffused could help rescuers to understand the impact of hazards and act on emergency responses.

Works of Imran et al.(2015) proposed an interesting survey of computational methods to process social media messages during mass emergency. Additionally, authors examined a series of key problems ranging from the detection of events to the creation of actionable and useful summaries from the collected data in emergencies. The survey covers particularly various approaches used to extract situational awareness information from social media and highlight their advantages and issues.

Moreover, authors addressed challenges related to the processing of social media messages. These were classified into two categories: scalability and content. The scalability issue as explained by authors is related to: the size, the velocity and the redundancy of data. The content issues are related to: the informality, the heterogeneity of the messages and the variety of languages used.

Challenges related to event detection in a mass of data were addressed in this paper too. Those are related to three points according to the authors: the inadequate spatial information due to the no GPS information, the mundane events that make noise on the processing algorithm while working and the description of events.

7. DISCUSSION

In this section, we will discuss the challenges and open issues related to the use of WSNs, IoT and social media in the search and rescue activities.

7.1. WSNs in Search and Rescue Applications, Challenges and Open Issues

Adding a wireless sensor network to a disaster area can assist rescue operations by locating survivors, identifying risky areas and making the rescuer aware of the overall situation. This can increase the efficiency of rescue operations and ensure the safety of the rescuer. However, this wireless sensor network has several challenges related to the hardware of sensors used, to the energy consumption, to routing protocols, the degree of autonomy and to the quality of services.

First, challenges related to the hardware: in search and rescue system, sensors are running under an extreme dangerous environment, which has water, high temperature, fire, falling debris, toxic liquid, dangerous gas, radioactive substances and biochemical materials. All these elements represent a harsh environment for sensors, which could abyss sensors, and so they stop working. Considering this challenging environment, sensors should be protected and packaged before using them.

Second, there are some challenges related to the energy consumption, the main energy consumed by the sensor is related to communication and data processing required by tasks allocated to the sensor. Some of the important tasks that have to be done by sensors in a search and rescue system is the inspection of the environment and the track of victims and robots. These, require stationary sensors to do a timely discovery of the change in the environment and of the location of the mobile tracked elements. To have an energy-efficient discovery schemes we need to reduce the energy consumption by stopping sensors from working from time to time and by reducing the data exchange. This should be done without missing contacts with the tracked mobile elements and with being aware of any change in the disaster environment in real time. However, it is highly dangerous to tolerate missing the change in the environment for example in the case of fire, this could invade safe areas in any time and the rescuers should be rapidly notified about these changes. For the timely discovery, most of the solutions proposed in the literature embed a simple periodic wakeup scheme with an active period (Jain et al. (2006)). However, changes in disaster place are unpredictable so the periodic wakeup time of sensors should be not only fixed dependent on the mobility pattern of robots and victims but also dependent on the change of the environment.

Another important task that could consumes energy is related to the data transfer protocol, having an efficient protocol that communicate information from sensors to the mobile nodes and to the rescue center help in the control of the energy consumption. Also using encoding and aggregation methods may reduce the size of data transferred without losing the information about the environment. Additionally, avoiding overloading nodes around the sink with data and having load balance between nodes may be helpful in this situation too.

Third, challenges related to routing protocols: the main goal of routing protocol is to find a way to successfully and efficiently deliver packets in the network. This is a challenging task since the network architecture is not fixed and change in an unpredictable way due to the nature of the disaster area. Additionally, obstacles that are frequent and could appear at any time in the disaster place create holes in the network coverage and make some areas isolated from the network, as a consequence, data will be sent through longer paths. The second issue that should be considered in routing protocols is related to the fact that some data need to pass by some predefined sensors to reach the rescue center. Therefore, these sensors die off faster than others in the network. To fix this problem the architecture should prevent additional sensors to put them in those areas.

Forth, challenges related to the quality of service, it is hard and a bit complicated to have a quality of service (Qos) in a disaster area because of the dynamic network. This QoS could be related to the identification of efficient and reliable path to move through the robot or to upload the data to the rescue center. The selected path should meet the QoS required by the application, for example in fire accident the selected path should be as far as possible from fire. The challenge is to define the most suitable path with exchanging the less data possible. Also having alternative paths for load balancing between nodes and QoS.

Fifth, challenges related to the coverage hole problem. Some solutions seen previously in this article propose to route data from the safest path or from the shortest path and guide the robot to move through this road. This overloads the area with the data exchange and as a consequence, it may create a hole in the network coverage. Some holes in the network coverage creates isolated areas, which means areas that are not connected to the rescue center and to the rest of the network. Supposing that there are victims in this isolated area whereby robots and rescuers will not be aware of their existence with probable fatal consequences. Additionally, robots that wait in this area to receive a call for rescue will not be informed about call for rescues and could not receive order from the rescue center.

To treat the hole coverage problem, a basic solution is to use mobile robots that move to cover the uncovered areas. Solution that might be more interesting to the case of WSRN is to equip robots with sensors to deploy them in the uncovered areas. For these, we have to decide how to detect the hole, it is possible to detect it by sensors or we could give a robot the mission to go around the network looking for any holes. After detecting the hole, sensors route a message report to the robot and this latter deploys sensors in the area. We think that the hole coverage problem could be really frequent in search and rescue system and there is many open issues in this research field that could be treated in the future such us: time that the robot take to redeploy a sensor, reparation cost (energy consumed and number of messages exchanged) and predicting the existence of holes in the network.

Sixth, there are other challenges related to the rapid attraction of robots to a target without conflicts. To the orientation of robots to victims or exits with lower cost. Also to the optimization of the number of expensive robots used in the search and rescue activities.

7.2. Open Issues in IoT for Search and Rescue Applications

At nowadays, many devices are adopting IoT technology, which are helpful for search and rescue activities especially for data collection during emergencies. Even that IoT has achieved a high degree of maturity we believe that it still can be improved.

One important issue concerning the use of IoT is related to the security. Indeed, false alarm and falsified data in search and rescue applications are not tolerable. A hacker can easily falsifies alarms and data sent to the rescue center. For these reasons, a good security protocol should be used to protect data exchanged in the network. Another issue is related to the use of advanced algorithms and machine-learning techniques to convert data collected from the IoT devices into knowledge and valuable information. One other issue is related to the autonomy of the IoT devices. These, are supposed to have a good autonomy to make decisions related to the search and rescue situation. We believe that more works should be done to solve the above issues.

8. CONCLUSION

In this paper, we first reviewed robotic search and rescue systems, by defining these latter and their environments. Additionally, we presented challenges related to these latter and tasks that a robot or a multi-robot system should execute to fulfil the search and rescue activities. Then, we exposed a system that integrates WSN with robotic system in order to better accomplish rescue tasks. We presented the advantages that bring the WSN to robotic systems in search and rescue activities. In addition, we cited tasks and missions that are achieved in a better way with a cooperation of WSN and robots. Furthermore, we exposed the remarkable researches that include this cooperation and the remaining open issues. We then considered several issues that WSNs brings to robotics search and rescue systems and the open research challenges.

Recent research shows interest to the use of a cooperation of wireless sensor networks and robots for search and rescue due to the reliability and strength that give WSNs to robotic systems, which can replace the human presence. Since this field is recently adopted to the search and rescue systems, there is many areas, which continue to be studied. For example, solutions that propose to save energy of the radio by sampling, processing and aggregating data, if we adopt them to search and rescue activities, it could be a big step forward. This gives long life batteries for both robots and sensors, which avoids a redeployment of sensors and the change of batteries of robots on the middle of the search and rescue activity due to holes on the coverage and to avoid the risk of losing the tracked robots or victims. Additionally, an adoption of periodic wakeup time protocol to save energy for search and rescue environment is needed. This should have a wakeup time that is fixed according to the mobility pattern of the robot, victims and the predictable changes in the environment and the topology of the network. Furthermore, it could be interesting to adopt the energy-efficient real time protocols; these keep giving the quality of service required for real time communication between the robot and the rescue center without losing energy. Search and rescue systems based on the cooperation of wireless sensor network and robot still need to be more explored, and improved, which gives to researchers many research areas to be discussed.

REFERENCES

Ai, C., Li, F. H., & Zhang, K. (2017). Detecting isolate safe areas in wireless sensor monitoring systems. *Tsinghua Science and Technology*, *22*(4), 427–443. doi:10.23919/TST.2017.7986945

Allali, S., & Benchaiba, M. (2017). Safe route guidance of rescue robots and agents based on hazard areas dissemination. *2017 International Conference on Mathematics and Information Technology (ICMIT)*. doi:10.1109/mathit.2017.8259692

Allali, S., Benchaïba, M., Ouzzani, F., & Menouar, H. (2018). No-collision grid based broadcast scheme and ant colony system with victim lifetime window for navigating robot in first aid applications. *Ad Hoc Networks*, *68*, 85–93. doi:10.1016/j.adhoc.2017.10.006

Allali, S., Menouar, H., & Benchaïba, M. (2016). Grid architecture for lightweight WSN-based area monitoring and alerts dissemination. *2016 International Symposium on Networks, Computers and Communications (ISNCC)*. 10.1109/ISNCC.2016.7746110

Asakura, K., Takikawa, H., Oishi, J., & Watanabe, T. (2010). An ad-hoc unicursal protocol for stable and long-lived communication systems in disaster situations. *International Journal of Knowledge and Web Intelligence*, *1*(3/4), 154. doi:10.1504/IJKWI.2010.034185

Chen, J., Yang, Y., & Wei, L. (2010). Research on the Approach of Task Decomposition in Soccer Robot System. *2010 International Conference on Digital Manufacturing & Automation (ICDMA)*. 10.1109/ICDMA.2010.33

Chen, Z., Li, Z., Li, J., Chen, J., & Liu, Y. (2011). Quasi Real-Time Evaluation System for Seismic Disaster Based on Internet of Things. *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing (iThings/CPSCom)*. doi:10.1109/ithings/cpscom.2011.112

Gelenbe, E., & Wu, F. (2012). Large scale simulation for human evacuation and rescue. *Computers & Mathematics with Applications (Oxford, England)*, *64*(12), 3869–3880. doi:10.1016/j.camwa.2012.03.056

Imran, M., Castillo, C., Diaz, F., & Vieweg, S. (2015). Processing Social Media Messages in Mass Emergency. *ACM Computing Surveys*, *47*(4), 1–38. doi:10.1145/2771588

Jain, S., Shah, R. C., Brunette, W., Borriello, G., & Roy, S. (2006). Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks. *Mobile Networks and Applications*, *11*(3), 327–339. doi:10.100711036-006-5186-9

Jiang, J. R., Lai, Y. L., & Deng, F. C. (2011). Mobile Robot Coordination and navigation with directional antennas in positionless Wireless Sensor Networks. *International Journal of Ad Hoc and Ubiquitous Computing*, *7*(4), 272. doi:10.1504/IJAHUC.2011.040775

Kerman, M. C., Jiang, W., Blumberg, A. F., & Buttrey, S. E. (2009). *Event detection challenges, methods, and applications in natural and artificial systems*. Lockheed Martin MS2.

Ko, A., Lau, H. Y., & Lee, N. M. (2009). Ais based distributed wireless sensor network for mobile search and rescue robot tracking. *Journal of Mathematical Modelling and Algorithms*, *8*(2), 227–242. doi:10.100710852-009-9105-5

Li, N., Sun, M., Bi, Z., Su, Z., & Wang, C. (2014). A new methodology to support group decision-making for IoT-based emergency response systems. *Information Systems Frontiers*, *16*(5), 953–977. doi:10.100710796-013-9407-z

Liu, J., Rojas-Cessa, R., & Dong, Z. (2016). Sensing, calculating, and disseminating evacuating routes during an indoor fire using a sensor and diffusion network. *2016 IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC).* doi:10.1109/icnsc.2016.7479014

Liu, J., Wang, Y., Li, B., & Ma, S. (2007). Current research, key performances and future development of search and rescue robots. Frontiers of Mechanical Engineering. *Frontiers of Mechanical Engineering*, *2*(4), 404–416. doi:10.100711465-007-0070-2

Loukas, G., Timotheou, S., & Gelenbe, E. (2008). Robotic wireless network connection of civilians for emergency response operations. *2008 23rd International Symposium on Computer and Information Sciences (ISCIS).* doi:10.1109/iscis.2008.4717943

Milford, M., & George, A. (2013). Featureless Visual Processing for SLAM in Changing Outdoor Environments. *Springer Tracts in Advanced Robotics Field and Service Robotics*, 569-583. doi:10.1007/978-3-642-40686-7_38

Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., & Erkmen, A. M. (2008). Search and Rescue Robotics. Springer Handbook of Robotics, 1151-1173. doi:10.1007/978-3-540-30301-5_51

Nagatani, K., Otake, K., & Yoshida, K. (2013). Three-Dimensional Thermography Mapping for Mobile Rescue Robots. *Springer Tracts in Advanced Robotics Field and Service Robotics,* 49-63. doi:10.1007/978-3-642-40686-7_4

Nasridinov, A., Ihm, S., Jeong, Y., & Park, Y. (2014). Event Detection in Wireless Sensor Networks: Survey and Challenges. *Lecture Notes in Electrical Engineering Mobile, Ubiquitous, and Intelligent Computing (MUSIC),* 585-590. doi:10.1007/978-3-642-40675-1_87

Nerurkar, E. D., Zhou, K. X., & Roumeliotis, S. I. (2011). A hybrid estimation framework for cooperative localization under communication constraints. *International Conference on Intelligent Robots and Systems 2011 (IEEE/RSJ).* 10.1109/IROS.2011.6094684

Okada, Y., Nagatani, K., Yoshida, K., Tadokoro, S., Yoshida, T., & Koyanagi, E. (2011). Shared autonomy system for tracked vehicles on rough terrain based on continuous three-dimensional terrain scanning. *Journal of Field Robotics*, *28*(6), 875–893. doi:10.1002/rob.20416

Teixeira, T., Dublon, G., & Savvides, A. (2010). A survey of human-sensing: Methods for detecting presence, count, location, track, and identity. *ACM Computing Surveys*, *5*(1), 59–69.

Teixeira, T., Dublon, G., & Savvides, A. (2010). A Survey of Human-Sensing: Methods for Detecting Presence. *Count, Location, Track, and Identity*, *3*(10), 133.

Timotheou, S., & Loukas, G. (2009). Autonomous networked robots for the establishment of wireless communication in uncertain emergency response scenarios. *Proceedings of the 2009 ACM Symposium on Applied Computing (SAC '09)*. 10.1145/1529282.1529542

Tuna, G., Gungor, V. C., & Gulez, K. (2014). An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters. *Ad Hoc Networks*, *13*, 54–68. doi:10.1016/j.adhoc.2012.06.006

Wang, K. M., & Hui, L. (2017). Effectiveness evaluation of Internet of Things-aided firefighting by simulation. *The Journal of Supercomputing*. doi:10.100711227-017-2098-3

Yan, Z., Jouandeau, N., & Ali, A. (2013). A Survey and Analysis of Multi-Robot Coordination. *International Journal of Advanced Robotic Systems*, *1*. doi:10.5772/57313

Yuan, J., Tang, S., Wang, C., De, D., Li, X., Song, W., & Chen, G. (2011). A real-time rescue system: Towards practical implementation of robotic sensor network. *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. doi:10.1109ahcn.2011.5984930

Chapter 8
A Current Review of Human Factors and Ergonomic Intervention With Exoskeletons

Thomas M. Schnieders
Iowa State University, USA

Richard T. Stone
Iowa State University, USA

ABSTRACT

Research and development of exoskeletons began as early as the 1960s. Recent advancement in technology has spurred a further research into the field specifically at rehabilitation and human performance augmentation. Human performance augmenting exoskeletons find use in the military, emergency services, industrial and space applications, and training. Rehabilitation exoskeletons assist in posture support and replacing lost function. Exoskeleton research is broadly broken up in this chapter by anthropometry: lower body, upper body, and extremities. The development for various anthropometry has their own unique set of challenges. This chapter provides a brief history, discusses current trends in research, looks at some of the technology involved in development, the potential benefits of using exoskeletons, and looks at the possible future improvements in research.

INTRODUCTION

This chapter provides a brief history of exoskeleton development from a human factors and ergonomic intervention standpoint. It discusses the current research with respect to the innate human-machine interface and the incorporation of exoskeletons for ergonomic intervention. Some novel exoskeletons based on their anatomical

DOI: 10.4018/978-1-5225-5276-5.ch008

categories of lower body, upper body, extremities (hands/feet), and full body exoskeletons will be discussed. The chapter will conclude by covering the benefits of exoskeletons in rehabilitation, industrial applications, and military applications, as well as discuss some of the issues faced when designing exoskeletons.

It is imperative when performing research with exoskeletons to clearly define the difference between an exoskeleton and an orthotic because the two terms often overlap in the media as well as in the scientific literature. The term orthotic, or orthosis (plural orthoses) is generally a mechanical device applied externally to the body.

Similarly, an exoskeleton is also a mechanical device applied externally to the body but whose external mechanical structure has joints that anatomically match the human body. Typically, this mechanical structure will share physical contact with the user, will enable a direct or indirect transfer of mechanical power, and has either active or passive actuation (Pons, J.L., Moreno, J.C., Brunetti, F.J., and Rocon, E., 2007). A prosthetic is a device that substitutes a missing body part (Sansoni, S., Wodehouse, A., and Buis, A., 2014).

Hugh Herr defines exoskeletons and orthoses as follows: "The term 'exoskeleton' is used to describe a device that augments the performance of an able-bodied wearer, whereas the term 'orthosis' is typically used to describe a device that is used to assist a person with a limb pathology" (Herr, H., 2009).

OVERVIEW OF EXOSKELETONS

The field of exoskeleton research, design, and manufacturing is broad and expansive. The earliest work began in the early 1960's with the Defense Department's interest in a man-amplifier. This man-amplifier was a "powered suit of armor" that can augment a soldier's lifting and carrying ability (Kazerooni, H., Steger, R., and Huang, L., 2006).

General Electric's Hardiman, developed by Ralph Mosher from the early1960s to 1971, is widely considered the first exoskeleton device. This hydraulic and electrical body suit made carrying 250-pound seem as light as lifting 10-pounds. Ultimately, the concept of the Hardiman was well received but determined too heavy and bulky for standard military use. Not only was the exoskeleton heavy (weighing in at 1500 pounds), but it was also extremely slow (2.5ft/sec) and had uncontrolled, violent movements (Ali, H., 2014).

The United States Air Force commissioned the study of a man-amplifier from the Cornell Aeronautical Laboratory in 1962. The results of their study on this master-slave robotic system showed that exoskeletons, even those with fewer degrees of freedom (DoF) than the human body, was capable of completing most desired tasks (Mizen, 1965). However, the master-slave systems currently being used were deemed impractical. The devices were overly complex and had difficulty sensing human intent making many tasks difficult to complete (Kazerooni, H., Steger, R., and Huang, L., 2006).

Throughout the 1960's and 1970's, the University of Belgrade developed several types of exoskeletons that had limited predefined motion to aid paraplegics. They were met with limited success but the balancing algorithms that were developed are still in use in many current day bipedal robots (Vukobratovic, M., Ciric, V., and Hristic, D., 1972).

USES AND MARKETS

In today's market, exoskeletons are broadly placed into two primary functions: (1) rehabilitation and (2) human performance augmentation. The recent surge into exoskeleton design, research, and manufacturing has led to their use into other fields like firefighting, sports, and law enforcement. The number of journal and conference papers covering exoskeletons each year has grown exponentially from just over 100 in 1964 to well over 7,000 in 2016. The number of publications on the term 'exoskeleton' is most likely skewed for years prior to the 1990's where the term was more often associated with the traditional sense of the word – that is, a rigid external covering for some invertebrate animals like arthropods.

Rocon (Rocon, E. et al. 2007) and Harwin (Harwin, W., Leiber, L., Austwick, G., and Dislis, C., 1998) classify rehabilitation robotics (and by extension rehabilitation exoskeletons) into three categories:

1. Posture support mechanisms
2. Rehabilitation mechanisms
3. Robots [and exoskeletons] to assist or replace body functions

Human performance augmentation exoskeletons are classified as exoskeletons designed to enhance, or augment, the capabilities of *healthy* people. Applications vary from fatigue reduction and lifting assistance to military focused use of increasing soldiers' endurance and load carrying capacity. Other examples of human performance augmentation classified exoskeletons include emergency services, construction, material handling (Brown, M., Tsagarakis, N., and Caldwell, D.G., 2003), and weapons training (Schnieders, T.M., Stone, R.T., Oviatt, T., and Danford-Klein, E., 2017) as well as any application that requires heavy lifting or heavy gear in rough terrain that is otherwise impassable by vehicles.

LOWER BODY EXOSKELETONS

Exoskeletons that are utilized in the lower body are primarily comprised of hip joint, knee joint, and ankle joints. Among the many different challenges involved in developing an exoskeleton for lower body are the interface between human and the exoskeleton, portable energy sources, controls, and actuators. The lower body exoskeletons can be divided into two types based on application: rehabilitation and enhancement capabilities of healthy human beings.

Most lower body exoskeleton robots were first designed to assist soldiers in supporting equipment. These wearable lower body exoskeletons can significantly reduce the oxygen consumption of soldiers and support energy for walking, running and help movement and operational capability of the soldiers (Yuan, P., Wang, T., Ma, F., and Goug, M., 2014). It is important to understand the biomechanics of humans to develop designs of exoskeletons and active orthoses for the lower limbs (Dollar, M.A. and Herr, H., 2008).

The Berkeley Lower Extremity Exoskeleton, or BLEEX, is the first energetically autonomous robotic exoskeleton that was successfully demonstrated to provide the operator with the ability to carry significant loads with minimal effort over any type of terrain. BLEEX has four critical features: (1) A novel control scheme, (2) high-powered compact power supplies – hydraulic and electric actuations that have been designed to power BLEEX, (3) special communication protocol and electronics, and (4) design architecture to decrease the complexity and power consumption (Chu, A., Kazerooni, H., and Zoss, A., 2005) (Zoss, A.B., Kazerooni, h., and Chua, A., 2006).

The BLEEX leg kinematics are similar to the human leg kinematics (anthropomorphic architecture). In general, BLEEX has ankle, knee, and hip joints like a person. The ankle has three DoF, just like a human ankle joint. However, only the dorsi/plantar flexion joint is aligned with the human joint. The knee has only one DoF (rotary) while the human knee has three DoF (rotary and sliding). This allows BLEEX to perform only basic flexion and extension. The hip has, like the human, three DoF. However, only the BLEEX hip flexion and extension joint is aligned with the human hip. BLEEX's final DoF is compliancy built into the front of the foot to yield similar flexibility to human toe joints (Zoss, A. and Kazerooni, H., 2006).

The BLEEX enables its wearer to carry a heavy load. It was first presented by U.C. Berkeley's Human Engineering and Robotics Laboratory supported by the Defense Advance Research Project Agency (DARPA) (Dollar, M.A. and Herr, H., 2008). The BLEEX seeks to supplement the intelligence and sensory systems of a human with significant strength and endurance with a pair of wearable robotic legs that offers a payload capacity (Chu, A., Kazerooni, H., and Zoss, A., 2005).

The HAL-5 system was developed from HAL-3. It is composed of three main parts: (1) exoskeleton and actuator, (2) controller, and (3) sensor (Uemura, M., Kamaoka, K., and Kawamura, K., 2006). Upgraded from HAL-3, the HAL-5 system utilizes a number of sensing modalities to control the EMG electrodes placed below the hip and above the knee on both the front and back sides of the user's body by connecting measurements, sensors, and the accelerometer (Herr, H., 2009) (Guizzo, E. and Goldstein, H., 2005).

In other words, the HAL-5 was updated from previous versions to include upper-body limbs, lighter and more compact power units, longer battery life, and a better body shape to fit humans more easily. This suit also includes two control systems— voluntary control and autonomous control (Guizzo, E. and Goldstein, H., 2005). Therefore, this version of the HAL exoskeleton is designed not only to help disabled patients in hospitals and the elderly, but also to support workers with demanding physical jobs, such as disaster rescue or construction (Guizzo, E. and Goldstein H., 2005).

Sarcos, an engineering and robotics firm, first developed the XOS2, a second-generation robotics suit, in 2006 after receiving a grant from DARPA. Sarcos was purchased by Raytheon in 2007.

The wearable suit enables the user to enhance human strength and agility, supports the soldier's capabilities for movement with power, and lift heavy objects (Raytheon, 2014). The XOS2, updated from a first-generation robotic suit—the XOS1 model, has the capability for a weight overload on one foot by using powered limbs. This exoskeleton can punch a bag, climb stairs, and kick a ball (Raytheon, 2014). Although these dynamic functions of the suit have been developed, an energy problem with the suit has not yet been resolved. It is limited due to a low battery density (Yuan, P., Wang, T., Ma, F., and Gong, M., 2014).

The XOS2 is a whole-body exoskeleton and does not entirely imitate the shape of humans with a much larger frame (Yuan, P., Wang, T., Ma, F., and Gong, M., 2014).

For military usage, the XOS2 needs to be reduced in size and have a simpler, more anatomic structure. The XOS2 utilizes several different sensors to minimize the amount of assistance the soldier receives. This allows for less effort by the soldier because the exoskeleton supports a large portion of the load (Yuan, P., Wang, T., Ma, F., and Gong, M., 2014)

The ReWalk, a wearable robotic exoskeleton, (Argo Medical Technologies Ltd., Yokneam Ilit, 20692, Israel) is one of the most unique exoskeletal robotic technology products that utilizes people's movements to control externally powered walking (Reinkensmeyer, D.J., Dewald, J.P.A., and Rymer, W.Z., 1999). The ReWalk supports powered hip and knee motion to enable individuals with a spinal cord injury (SCI) to stand upright and walk (ReWalk, 2014).

The design system allows independent, controlled walking and standing while simulating the natural gait patterns of the legs (Zelig, G. et al., 2012). The pilot study found that the ReWalk is safe and tolerated well. There were no significant complications related to cardiovascular stresses, fatigue, excessive skin pressure, and muscle pain (Zeilig, G., Weingarden, H., Zwecker, M., Dudkiewicz, I., Bloch, A., and Esquenazi, A., 2012).

In the study, eight participants successfully reached different levels of walking. Although the devices have significant potential physiological benefits, they still have not attained proficiency to be a functional daily use device. One of the major issues is that functional use of the commercially available ambulation devices for paraplegics is the high-energy demands imposed.

Developed by the Wyss Institute, the Soft Exosuit consists of a combination of sensors, such as a hyperelastic strain sensor and sensors around the wearer's hips, calves, and ankles secured by straps (Soft Exosuit, 2014). This Soft Exosuit has strong commercial potential for helping spinal-cord injury patients walk or helping soldiers carry heavy loads (Soft Exosuit, 2014).

The main benefit of the Soft Exosuit is its extremely light design due to the soft material. The soft flexible materials cannot only interface with the wearer but also provide a flexible structure so assistive torques can be applied to the biological joints (Soft Exosuit, 2014).

The wearer's bone structure must sustain all the compressive forces normally encountered by the body plus the forces generated (Asbeck, A.T., Dyer, R.J., Larusson, A.F., and Walsh, C.J., 2013). Therefore, the Soft Exosuit, as a potential tool, cannot only help physical workers with hard tasks and support gait but also assist in rehabilitation and protection from injury or spinal cords from impairment from heavy physical activity (Asbeck, A.T., Dyer, R.J., Larusson, A.F., and Walsh, C.J., 2013).

Focused on Low impedance – the RoboKnee, a prototype exoskeleton, presents low impedance to the wearer and has a natural interface. To achieve transparency, the exoskeleton must successfully perform the following functions: determine the user's intent, apply forces when and where appropriate and present low impedance, i.e. "get out of the way."

A series elastic actuator is used to achieve low impedance and is attached to the lower and upper part of the leg so that it provides torque across knee providing one DoF. User intent is determined through the knee joint angle and ground reaction forces (Pons, J.L., Moreno, J.C., Brunetti, F. J., and Rocon, E., 2007).

The RoboKnee allows the wearer to climb stairs and perform deep knee bends while carrying a significant load in a backpack. The device provides most of the energy required to work against gravity while the user stays in control, deciding when and where to walk, as well as providing balance and control (Pratt, J.E., Krupp, B.T., and Morse, C.J., 2014).

Due to low energy density of batteries, the RoboKnee does not yet achieve the long-life requirement. While it is very comfortable to use, the current implementation is somewhat difficult to don and doff. While the RoboKnee enhances strength and endurance, it was not designed for enhancing the user's speed, and, in fact, it restricts the user from running (Pratt, J.E., Krupp, B.T., and Morse, C.J., 2014). A recommendation from authors was to develop exoskeleton that incorporates other joints than just the knee (Pratt, J.E., Krupp, B.T., and Morse, C.J., 2014).

Lower exoskeleton robots should have the following characteristics: (1) light weight actuation and efficient transmission; (2) ability to maintain power, actuation, and other subsystems, off the shelf components do not meet the low weight, high efficiency, and other criteria needed to accomplish their design objective; and 3) examine quantitative performance results for exoskeleton devices that reportedly improve human locomotion.

To achieve the above characteristics, lower exoskeleton robots should develop computing, sensing, and control without pervasive application. Therefore, matching the structure of the exoskeleton to the wearer is one of fundamental factors: (1) alignment between joints of robot and the wearer; (2) segment running or jumping ability; (3) safety of human operator; (4) interfacing exoskeleton or active orthoses to the human body naturally.

UPPER BODY EXOSKELETONS

Upper body exoskeletons have many similar complications to that of the lower body but have their own additional slew of challenges that stem from their more varied types of upper body applications. While the primary function for most lower body exoskeletons is to bear weight and transport loads, "the main function of the arm is to position the hand for functional activities" (Rocon, E. et al., 2007). In addition to the more complicated functions being performed, the upper limb joint anatomy is complex. For example, the shoulder is located near three bones (clavicle, scapula, and humerus), and has four articulations. The calculations have involved dynamics and irregular centers of rotation (Gopura, R.A.R.C. et al., 2009).

A lot of research and development was done for using exoskeletons in the medical and rehabilitation field. While ABLE, developed at CEA-LIST Interactive Robotics Unit, was designed to have multiple purposes, its first application was in a rehabilitation project.

Further applications for industry and professional fields are in view such as intuitive telerobotics, as a haptic device for Virtual Reality (VR), and for sports training. ABLE used a circular guide for the shoulder joint, which solved the problem of singularity. The exoskeleton used a screw cable system and was able to be integrated with the motor power transmission of another robotic system without modification. The ABLE-7 axis model has lighter weight and a 3-axis open forearm-wrist, which can be used for teleoperation and virtual reality (Garrec, P., Friconneau, J.P., Measson, Y., and Perrot, Y., 2008).

ARMin is used for the rehabilitation purpose of the arm, which has four DoF and two passive DoF enabling elbow flexion/extension and shoulder rotations. It was installed with multiple sensors for position, force, and torque, so that this robotic system can combine the cooperation and motivation of the patients into the therapy activities and give the support to the patients based on their needs.

The development of ARMin is in prototyping and additional supervision from therapist needed for safety. Special mechanical design can be seen in ARMin, which includes customized module for upper arm rotation enabling small friction during rotation and patient comfort for wearing the device (Miller, C., Ruiz-Torres, R., Stienen, A. H. A., Dewald, J. P. A., 2009).

ARMin II, the second iteration, is a 7 DoF robotic system designed for therapy purposes. It can adapt to different patients' sizes with adjustments of five parts and be optimized for combining user cooperation with control strategies. The ARMin II is under evaluation and testing for further improvements (Mihelj, M., Nef, T., and Riener, R., 2007).

The Cable-Actuated Dexterous Exoskeleton for Neurorehabilitation (CADEN-7) is an anthropometric 7 DoF powered exoskeleton system that offered several advantages over other designs at the time of its development. "The anthropomorphic nature of the joints combined with negligible backlash in seven force reflecting articulations sets the CADEN 7 apart from other arms in the field (Perry, J and Rosen, J., 2006)." The CADEN-7 features low-backlash gearing, back drivable transmissions, low-inertia links, high stiffness transmissions, open mechanical human machine interfaces (mHMI's), and a range of motion (ROM) representing 99 percent of a human physiological ROM (Perry, J. and Rosen, J., 2006).

CADEN-7 was used in the development of myoprocessors for the upper limb based on the Hill phenomenological muscle model. Genetic algorithms were used to optimize the internal parameters of the myoprocessors using an experimental database that provided inputs to the model. Research results indicated high correlation between joint moment predictions of the model and the measured data, suggesting that the myoprocessor was sufficiently robust for further integration into exoskeleton control systems (Perry, J. and Rosen, J., 2006).

Cable-driven arm exoskeleton (CAREX) has lighter weight compared to a traditional rigid exoskeleton. Cables are used to move human upper body segments, which are powered by motors and guided by cuffs. CAREX can provide push and pull with predefined force in required direction during rehabilitation trainings (Mao, Y. and Agrawal, S.K., 2012).

The current Mirror Image Movement Enabler (MIME) robotic device was the development result from the continuous work from 1998. Early research indicated that bilateral training was more effective than unilateral training when using similar movements. A robotic controller (PUMA 560) was used to manipulate the forces needed, which therapists use for the normal therapy training.

The movements assisted by robot can be classified as 4 different types: passive, active-assisted, active-constrained, and bilateral. During passive training, the robot moves the human arm to reach the target with a defined path without human effort. During active-assisted training, humans initiated the movement using force and collaborated with the robotic device to reach the target. During active-constrained training, desired movements were defined by the robot and maximum effort of the human was needed to reach the target. During bilateral training, the target arm was assisted by the robot to do the same movement as the contralateral arm. In this study, Fugl-Meyer and EMG data were collected and analyzed for rating the improvement of the participants and the muscle engagement during the training (Lum, P.S. et al., 2006).

The SUEFUL-7 is 7 DoF upper-limb motion assist exoskeleton robot used to test electromyographic (EMG) control methods using neuro-fuzzy modifiers in assisting the motions of shoulder vertical and horizontal flexion/extension, shoulder internal/external rotation, elbow flexion/extension, forearm supination/pronation, wrist flexion/extension, and wrist radial/ulnar deviation of physically weak individuals. The use of the neuro-fuzzy modifiers allows impedance parameters to be adjusted in real time by considering the upper-limb posture and EMG activity levels (Gopura, R. A. R. C. et al., 2009).

WREX was designed and developed to assist children for daily living activities who do not have enough strength for their arms. WREX has 5 DoF and allows a wide range of naturalistic motions. The modified version for training stroke patients was called Therapy Wilmington Robotic Exoskeleton (T-WREX). The T-WREX contains an orthosis, a grip sensor and software, which enables a wide reach of the arm across the workspace, hand grip pressure detection and functional training movement simulation (Sanchez, R.J. et al., 2006).

The wearable orthosis for tremor assessment and suppression (WOTAS) provides a means of testing and validating non-grounded (i.e., wearable) control strategies for orthotic tremor suppression (Rocon, E. et al., 2007). Unlike most exoskeletons that seek to enhance intended muscle movement, the purpose of WOTAS is to dampen unintended movement, and it can operate in both active and passive damping modes. The control algorithm of WOTAS must distinguish between wanted and unwanted movement.

The Titan Arm is a lightweight upper body exoskeleton designed to closely mimic the human range of motion and assist weakened individuals regain mobility and independence. The Titan Arm provides 3 DOF with non-localized actuation and a ratchet based braking system that allows the Titan Arm to hold static loads without requiring force from the user (Beattie, E., McGill, N., Parotta, N., and Vladimirov, N., 2012). The Titan Arm carries most of its weight in the back-plate and can augment the user's lifting strength by up to 40 lbs. In addition, the system can provide real-time joint tracking which can be streamed to a computer for analysis.

Most upper limb rehabilitation systems have been developed for unilateral training, but the upper limb exoskeleton rehabilitation device (ULERD) can be used for bilateral training. The ULERD has three active DoF and four passive DoF. The ULERD incorporates a commercial haptic device known as Phantom Premium, as well as an inertia sensor known as MTx, to detect input signals from one arm which is held stationary. The output movement is performed by a wearable exoskeleton on the right arm (Song, Z. and Guo, S., 2011).

EXTREMITIES

Much of the literature for hand exoskeletons points towards their use in rehabilitation. However, there has also been work done looking at the use of hand exoskeletons as haptic interfaces for interaction with the virtual environments and extra vehicular activities in space. In addition, hand-based exoskeletons have been making advancements in rehabilitation.

The Hand Exoskeleton Rehabilitation Robot, or HEXORR, is an exoskeleton whose robot joints are aligned with anatomical joints of the human hand and provides direct control of these hand joints. HEXORR uses a low-friction gear train and electric motors. This combination allows for both position and torque control which is an advantage. Additional advantages which HEXORR provides is the psychologically accurate grasping patterns which is controlled with just two actuators as against some complex designs that use eighteen actuators. All these factors make HEXORR unique compared to other devices (Schabowksv, C. M., Godfrey, S. B., Holley, R. J., and Lum, P. S., 2010).

The use of EMG signals to control exoskeletons is becoming more commonplace, especially in paraplegics and quadriplegics (Au, A.T.C., Arthur, R. C., and Kirch, R.F., 2000)(Ito, K., Tsuji, T., Kato, A., and Ito, M., 1992)(Kirsh, R.F., and Au, A.T.C., 1997) (Kuribayashi, K., Shimizu, S., Okimura, K., and Tamiguchi, T., 1993). There are over 12,000 new cases of spinal cord injury per year (FSCIP, 2014) and "nearly half of these cases result in a loss of sensation or motion to the arms and hands" (Lucas, L., DiCicco, M., and Matsuoka, Y., 2004). The researchers at Carnegie Mellon University developed an effective EMG-based hand exoskeleton that enabled pinching movements in patients who lacked hand mobility with the use of a functional electrical stimulation system - a system that stimulates muscles that no longer receive signals from the central nervous system - and a low-profile lightweight exoskeleton that consisted of "an aluminum anchoring plate mounted to the back of the hand and three aluminum bands, one for each of the finger bands (Lucas, L., DiCicco, m., and Matsoka, Y., 2004)," in conjunction with steel cabling that runs along the front of each finger band, a pneumatic cylinder, and a mechanical linkage mechanism.

An exoskeleton designed for upper arm rehabilitation and hand grasp training called the IntelliArm can control the user's shoulder, elbow, wrist, and finger with 8+2 DOF. The IntelliArm builds on the research of MIT MANUS, an upper body arm exoskeleton that assisted in arm reaching movements in post stroke rehabilitation (Hogan, N., Krebs, H.I., Sharon, A., and Charnaron, J., 1995) (Krebs, H.I. et al., 1997); Reinkensmeyer et al.'s Arm Guide robot, another upper body arm exoskeleton that was used to treat and evaluate post stroke patients by guiding their arm along a linear guide (Reinkensmeyer, D. J., Dewald, J. P. A., and Rymer, W. Z., 1999); and an industrial robot attached to a forearm splint called MIME, or a mirror image motion enabler, that assisted movement passively or actively (Burgar, C.G., Lum, P.S., Shor, P.C. and Machiel Van der Loos, H.F., 2000). The IntelliArm also built on the work of the ARMin, described earlier. The designs of the other rehabilitation robots that the IntelliArm built upon did not consider patients' hand posture. The researchers found that if they were to "ignore a proper control of the muscle tension of a subject's hand, the robot training may lower hand/fingers flexibility and potentially cause an abnormal muscle tone (Ren, Y., Park, H., and Zhang, L., 2009)."

A tendon driven exoskeleton that controls flexion of two degrees of freedom per finger was designed for physical therapy at the University of Salford (Sarakoglou, I., Tsagarakis, N. G., and Caldwell, D.G., 2004). A hand exoskeleton for the rehabilitation of stroke patients is the Rutgers Master II which actuates the user's fingers via four pneumatic pistons located inside the palm (Bonzit, M., Burdea, G., Popescu, G., Boizan, R., 2002).

Another hand exoskeleton designed for stroke patients is the Wrist/Finger Force Sensing module (WFFS) which is used during movements of the upper limb in chronic hemiparetic stroke patients. "The WFFS measures isometric flexion/extension forces generated by the wrist, fingers, and thumb during 3-D movements of the paretic upper limb (Mizen, N.J., 1965)." Unlike other hand exoskeletons, the WFFS can generalize 3-D movements of the hand in conjunction with the rest of the limb. This hand exoskeleton acts as a lightweight, portable, and rigid forearm orthosis of the ACT^{3D} robot. This allows for measurements of wrist and finger forces during any tasks that the ACT^{3D} normally performs.

One research focus is geared towards assisting astronauts in extravehicular activities or EVA. The current gloves used by NASA are less flexible than desired, requiring mechanical work to displace the glove and to hold the glove in any given position. This additional required work reduced EVA productivity and fatigues astronauts' hands. Work has been done to create a motorized hand exoskeleton with the ability to perform a power hand grasp and a precision finger grasp. The design consisted of a series of drivers, mechanical stops, sensor arrays, four bar linkages, DC motors, and cable driven cam systems. Human hands are particularly complex with over 25 degrees of freedom (Shields, B.L., Main, J.A., Peterson, S.W., and Strauss, A.M., 1997). The hand exoskeleton reduced the allotted degrees of freedom significantly leading to the system's primary shortcoming - the coupling of joints in the hand exoskeleton. The researchers found that if motion for one finger was attempted, the other fingers would also be forced to move, if only a little bit. Additionally, the sensor array would sometimes pick up hand motions that were not there causing undesired exoskeleton motion.

A robotic concept called Skil Mate was introduced to revitalize almost all human works skill on production sites introducing cooperation between human and robotic machines. This project was put into practice in August 1998. The aim of the project was to manufacture an exoskeletal structure to be worn by astronauts for EVA. It is designed to have no intelligence and memory but works synchronously with skilled workers. The exoskeletal structure covers the worker's arms, hands, fingers, body and legs (Umetani, Y. et al., 1999).

Much of the early literature for hand exoskeletons is geared towards their use as haptic feedback for virtual reality and augmented reality environments. VRLogic's CyberGrasp is a commercially available haptic interface for the hand that delivers a force feedback system to the fingers and hand. It utilizes pull cables with brakes on their distant end to restrict movement (VRLOGIC, 2014). A hand exoskeleton was developed at the Robotics Center-Ecole des Mines de Paris that can support bidirectional movement for two fingers. It is capable of four degrees of freedom for each finger, but can only control one at a time using a pull cable (Stergiopoulos, P., Fuchs, P., and Laurgaeau, C., 2003). Another hand exoskeleton developed for haptic feedback is the LRP Hand Master. It can support 14 bidirectional and actuated degrees of freedom (Tzafestas, C.S., 2003).

The reason for building the Knee Ankle Foot Orthosis, or KAFO, was to extend the pneumatically powered ankle orthosis concept to the knee and test its performance on healthy walkers. The KAFO was built with a unilateral powered knee-ankle-foot-orthosis with antagonistic pairs of artificial pneumatic muscles at both the ankle (i.e. plantar flexor and dorsiflexor) and the knee (i.e. Extensors and flexors) (Sawicki, G.S. and Ferris, D.P., 2009).

GAIT is an exoskeleton conceived as a compensation and evaluation system of pathological gait for application in real conditions as a combined assistance and assessment methodology of the problems affecting mobility in individuals with neuromotor disorders. Interaction with the human neuromotor system to assist locomotion requires adequate design of the components, both the biomechanical and functional aspects. Robotic exoskeletons conceived as an aid to mobility are designed to be used in numerous environments (Pons, J. L., Moreno, J. C., Brunetti, F. J., and Rocon, E., 2007).

FULL BODY EXOSKELETONS

The Berkeley Lower Extremity Exoskeleton, mentioned in the Lower Body portion of this document, is just the beginning work for the University of California, Berkeley. The researchers also plan to develop an upper body exoskeleton. After they are certain that both are capable of functioning independently, they will attempt to integrate the two systems (Kazerooni, H., 2005).

Ekso by Ekso Bionics is a primarily lower body exoskeleton for individual with any amount of lower extremity weakness or limb pathology to stand and walk. The exoskeleton uses battery powered motors to drive the legs when the exoskeleton's sensors pick up intended movement. The exoskeleton can provide natural gait and assists in gait training for patients who suffer from complete paralysis and who have minimal forearm strength (Ekso Bionics, 2014). The Ekso exoskeleton is considered a Class I medical device in the United States, a Class I medical device in Australia, and a Class IIa medical device in the European Union (Ekso Bionics, 2014).

Lockheed Martin's Human Universal Load Carrier (HULC) is a hydraulic-powered, titanium, anthropomorphic exoskeleton designed for military use. It can carry up to 200lbs, march at 3 mph, sprint at 10 mph, can travel 20 km on level terrain at 4 km per hour, and can be set to a long-range mode for extended 72-hour missions (HULC, 2014). The HULC weighs in at 53 lbs., is powered by lithium polymer batteries and can integrate with armor, heating and cooling systems, additional sensors, and other custom attachments.

The U.S. Government has officially sanctioned a full body exosuit for military use. The Tactical Assault Light Operator Suit, or TALOS, is the planned future of warfare. The US Army requested white papers from academia, industry, public labs, and any interested individuals on how to design and build TALOS (Singularity HUB, 2014). There hasn't been much released on this in development suit; however, there has been speculation that TALOS will feature an already designed exoskeleton at its core (Singularity HUB, 2014). The most likely candidates currently are Lockheed Martin's HULC and Raytheon's XOS 2.

TALOS, when fully completed, will be bulletproof, weaponized, able to monitor vitals, give wearer superhuman strength and perception, have layers of smart materials and sensors, and use wide-area networking and on-board computers to provide more substantial situational awareness (ARMY.MIL, 2014). The U.S. Army Research, Development and Engineering Command, known as RDECOM will be involved in every aspect of TALOS development.

THE TECHNOLOGY OF EXOSKELETONS

Exoskeletons are very complex machines that attempt to mimic and augment human motion. These machines typically augment human performance using a pump or other form of actuation. Sensing can be accomplished with EMG. Unlike the Pneumatic actuators in the section below, hydraulic pumps do not have a high degree of compliance. They are relatively rigid and can be made compliant only through use of complex strategies. The feature that distinguishes these artificial muscles is the way inflation and deformation is handled. At pressures of about 500kPa to 800kPa which is ideal, hydraulic operators suffer from a very bad power to weight ratio which makes them less attractive. Hydraulic operators are preferred in rare situations where the manipulator is powered underwater. In such cases, the weight problem is eliminated since the surrounding fluid is the same as the driving fluid (Daerden, F. and Lefeber, D., 2002).

The hydraulic system for an exoskeleton comprises of an oil tank, motor, oil pump, one-way valve, pressure sensor, servo valve and an actuating cylinder. The hydraulic system has an advantage that the current motion is being sensed by a sensor. The speed of the actuating cylinder can be controlled precisely by moving a spool of the servo valve, speed of the motor and the oil pump component (Patent 467004A, 2014). With hydraulic systems, high pressures can be used which aids to the reduction in size of the operating system. However, these systems are not fire proof and they are noted for leakage and high maintenance (Patent 467004A, 2014).

Electric motors are the favored option because they are simple to install, lightweight, small, generally silent, and offer a clean actuation system (Bouzit, M., Burdea, G., Popescu, G., and Boian, R., 2002).

Hydraulic actuators use pressurized fluid to transmit power to a joint. This actuator has a very large potential force and torque, but it is noisy and needs a complex apparatus to work (compressor, cooling system, etc.) (Bouzit, M., Burdea, G., Popescu, G., and Boian, R., 2002).

Pneumatic Actuators use pressurized gas to produce output force. In comparison to hydraulic actuators, pneumatic ones are lighter in weight and cheaper. Moreover, they offer a clean, non-flammable alternative actuation device (they also require a compressor or a rechargeable pressured tank). Artificial pneumatic muscles are referred to as McKibbon muscles (Bouzit, M., Burdea, G., Popescu, G., and Boian, R., 2002).

Pneumatic Actuators were largely used in automation segments of manufacturing. Lately this technology has been used by robotics as a main motion power source (Bouzit, M., Burdea, G., Popescu, G., Bonian, R., 2002). One of the most attractive features of the Pneumatic Actuator is the low weight trait. Its complaint behavior is also one of its favorable features. Compliance is due to the compressibility of air and can be controlled by adjusting the operating pressure. This is one of most important requisites when there is human-machine interaction. The compliance of the system guarantees a safe and easy interaction.

There are several types of pneumatic actuators: cylinders, bellows, pneumatic engines and pneumatic stepper motors. There is another kind called the Pneumatic Artificial Muscles (PAM). These contract on inflation since their core is very similar to a membrane which makes them extremely light-weight and spring like. Due to this nature they are widely used to power exoskeletons (Bouzit, M., Burdea, G., Popescu, G., and Boian, R., 2002).

Pneumatic Actuators are contractile and linear motion engines operated by gas pressure. Their core is made of a flexible membrane that is connected to the load on both ends by fittings and mechanical power. When the membrane is inflated it bulges and when deflated it gets squeezed. The force and motion thus generated are linear and unidirectional. The source of energy is usually air. The air pressure inside and outside the actuator is different and this aids to powering the actuator.

Magnetic brakes, as a passive technology, limit the motion by absorbing energy generated by the user. However, they are relatively heavy (Bouzit, M., Burdea, G., Popescu, G., and Boian, R., 2002).

Shape memory allows (SMA) are made from titanium-nickel (TiNi) metal and modify their shapes with a change in temperature. When heated, the wire or spring structure deforms producing very high forces. On the other hand, metal alloys can only generate small displacements (Bouzit, M., Burdea, G., Popescu, G., and Boian, R., 2002).

Piezoelectric motors convert vibrational motion into translational or rotational motion. When a piezoelectric material vibrates, frictional forces generate torque or force. Piezoelectrics are not widely used because they are difficult to manufacture and are quite expensive (Bouzit, M., Burdea, G., Popescu, G., and Boian, R., 2002).

Controls

- *Hand Held Controls*
    - ◦ Hand held pushbutton controllers are often used in conjunction with pneumatic muscles (Ferris, D.P. and Lewis, C.L., 2009).

- *Footswitch Control*
    - With this control scheme, the pressure in the pneumatic muscle is controlled based on the signal from a place in the subject's shoe (Ferris, D.P. and Lewis, C.L., 2009).
- *Proportional Myoelectric Control*
    - In this control scheme, the air pressure delivered to the artificial pneumatic muscle is directly proportional to the electromyographic (EMG) signal from a biological muscle (Ferris, D.P. and Lewis, C.L., 2009).
- *Passive Rehabilitation Controls*
    - Controls used in passive rehabilitation - without any effort from the wearer
        - Proportional Derivative (PD) controller (Lalami, M. et al., 2013)
        - Proportional Integral (PI) controller (Lalami, M. et al., 2013)
        - Proportional Integral Derivative (PID) controller (Lalami, M. et al., 2013)
        - Hybrid Assistive Limb (Combination of PD and EMG) (HAL-5) (Lalami, M. et al., 2013)
        - Exoskeleton Intelligently Communicating and Sensitive to Intention (EICOSI) controller (Lalami, M. et al., 2013)

Electromyography (EMG) can detect and receive signals from human skeletal muscle activity using electromyography. The muscle cells generate electrical signal during motion, which can be recorded and analyzed by EMG technology to study human biomechanics. This application was applied for a control method of exoskeletons using human intention for movements. EMG signals can represent the real-time muscle engagement information, which can enable manipulation of the exoskeleton in real time and simplify the control process. Gopura et al. (2009) combined Fuzzy-neuro with the EMG technique for the control process and used impedance as the intermediate controller to parameterize the real-time control based on postures and EMG signals. Problems of this method are that different people react different during motion as degrees of freedom increased the complexity of rules will increase as well (Gopura, R.A.R.C. et al., 2009).

Electro-rheological fluid can turn solid in less than a millisecond with the application of a voltage. This would provide resistance to motion on a healing joint and aims to help the muscle regain strength (Casolo, F., Cinquemani, S., and Coceta, M., 2008).

THE FUTURE OF EXOSKELTONS

Lo et al. (2012) stated that the exoskeleton training using in rehabilitation could potentially enable self-therapy activities without involvement of therapist, which can reduce the rehabilitation cost. Exoskeleton training could be flexible not limited to time and location, which can reduce schedule conflicts and provide a more frequent training. The cost associated with these problems can be reduced (Lo, H.S. and Xie, S.Q., 2012).

Rehabilitation improvement relies on intensity of training and patients' motivation. Recent studies on exoskeleton for rehabilitation indicate that the exoskeleton can provide trainings with different levels and more frequently compared to the traditional therapist training. Experimental results also show that the exoskeleton assisted trainings are effective for activities of daily living, which could benefit the stroke patients recover from neurological and orthopedic damages (Mihelj, M., Nef, T., and Rienner, R., 2007). Games are integrated into the exoskeleton training activities. Training processes are designed as games in order to provide patients with an entertaining experience, which can improve their motivation of therapy (Housman, S.J., Le, V., Rahman, T., Sanchez, R.J., and Reinkensmeyer, D.J., 2007) (Lo, H.S. and Xie, S., 2012).

Exoskeleton are used as human assistive devices in an industrial environment by reducing the load on human body, which extend the human capabilities. In virtual reality, the exoskeleton can be used as haptic device to allow human users to interact with virtual objects by parameterizing proper force based on their characteristics. Additionally, the exoskeleton served as a master device for manipulating control systems (Rosen, J., Perry, J.C., Manning, N., Burns, S., and Hannaford, B., 2005).

In order to enhance a soldier's capability and reduce soldier's workload, exoskeletons were developed to assist soldiers with better performance for heavy weapon carrying and firing (Winder, S.B. and Esposito, J.M, 2008). Among the most critical challenge lies in the design of a controller to allow natural movement of a highly articulate prosthetic with minimal ethical and physical invasion. For the foreseeable future, the first step is to determine a mapping from EMG patterns to muscle forces; this should be a primary research focus over the next few years. This method will allow individual finger movements coordinated with the hand, wrist, and elbow, unlike anything current prosthetics can accomplish. This will significantly increase the quality of life for the wearer and the utility of any prosthetic. Furthermore, perceiving and exploiting the intricacies of low-level neural signals will open the door for deeper understanding of cortical control and other methods tapping into spinal or peripheral nerves, thus jumpstarting the field of neuroprosthetics (Dellon, B. and Matsuoka, Y., 2007).

Actuator and power supply technologies still have limitations: current actuators are unable to provide both a high power-to-weight ratio and high bandwidth while modern power supplies have insufficient energy density (Lo, H.S. and Xie, S.Q., 2012). PMA has a high power-to-weight ratio but lack bandwidth while motors have sufficient bandwidth but have a poor power-to-weight ratio (Lo, H.S. and Xie, S.Q., 2012).

Current mobile exoskeleton robots rely on a lower limb exoskeleton to carry the weight of the actuators and power supply. Although this has been shown to a feasible approach with the recent success of the full body HAL-5 exoskeleton for assisting the elderly and physically weak, improvements on the weight and efficiency of the actuators and power supplies are needed to achieve better exoskeleton performance (Lo, H.S. and Xie, S.Q., 2012).

Another limitation is the singular configurations present in the exoskeletons 3 DOF shoulder complex which occurs when two rotary joints align with each other, resulting in the loss of 1 DOF. The current method used to address the problem merely shifts the configuration to an uncommon posture rather than eliminating the configuration from the upper limb workspace (Lo, H.S. and Xie, S.Q., 2012).

There is limited consideration of the interactions between the exoskeleton and the human user. The mechanical HRI location and interface area for optimal load transfer and comfort have not been considered in current exoskeletons (Lo, H.S. and Xie, S.Q. 2012).

The attachment locations of mechanical interfaces and EMG electrodes will inevitably vary each time the exoskeleton is worn. To enable better use of exoskeletons in practice, the device needs to be able to adapt to variations without long calibration downtimes.

WHAT CAN WE DO TO MAKE EXOSKELETONS BETTER?

There are few areas related to the mechanical design of exoskeletons that show promise and have largely been overlooked.

An improved understanding of walking and other movement may shed light on more effective exoskeleton leg architectures (Dollar, M.A. and Herr, H., 2008). Gait models based on actual machine elements that capture the major features of human locomotion may enhance the understanding of human leg morphology and control and lead to analogous improvements in the design of efficient, low-mass exoskeletons (Dollar, M.A. and Herr, H., 2008).

Investigation of non-anthropomorphic architectures may provide solutions to some of the problems associated with closely matching the structure of the exoskeleton to the wearer such as the need for close alignment between joints of the wearer and the exoskeleton (Dollar, M.A. and Herr, H., 2008). More research is required on recreational exoskeletons that augment running or with jumping ability (Dollar, M.A. and Herr, H., 2008).

Besides enabling technology and mechanical design there are a few issues related to the implementation of exoskeletons and active orthoses that needs further studying (Dollar, M.A., and Herr, H., 2008). Designing an exoskeleton with good mechanical strength, less weight, sufficient grip force, low power consumption, computational capability compatible to control scheme and high speed of operation (Singh, R.M. and Chatterji, S., 2012).

The design of structure is one area where an imaginative design may reduce lot of stress from weight constraint. The grip force and power consumption can be taken care by the proper choice of the actuators (Singh, R.M. and Chatterji, S., 2012).

The ideal requirements are materials for mechanical structure having mechanical strength, flexibility and weight like bone, the controller having computational capability, speed and adaptability like brain, actuator having high torque and flexibility like muscles, and the feedback elements having sensing capability like skin (Singh, R.M. and Chatterji, S., 2012).

EMG is a relatively new technology. It has a definite potential to be used as control signal for multifunction prosthesis. There is need to draw correlation between the physiological, physical factors and the EMG signal (Singh, R. M. and Chatterji, S., 2012). Advanced algorithms need to be developed to extract useful neural information (Singh, R. M. and Chatterji, S., 2012). One of the innovative aspects is the combined use of electroencephalogram (EEG) and electromyography (EMG) to relay information for controlling the lower-limb exoskeleton (Singh, R.M. and Chatterji, S., 2012).

WHAT ARE THE ISSUES FACED IN DESIGNING FOR EXOSKELETONS?

Current power supplies have insufficient energy density for truly mobile exoskeletons (Lo, H.S. and Xie, S.Q., 2012). Large, heavy power supplies limit portability and are one of the major factors limiting application of exoskeletons outside of clinical therapy (Lo, H.S. and Xie, S.Q., 2012) and other "grounded" applications. Some researchers have proposed interim solutions such as mounting upper body exoskeletons to powered wheelchairs (Kiguchi, K., Rahman, M, H., Sasaki, M., and Teramoto, K., 2008), but improvements on the weight and efficiency of power supplies are still needed to achieve better exoskeleton performance (Lo, H.S. and Xie, S.Q., 2012).

"A mechanism that synthesizes a human-type motion will necessarily also be complex, particularly from the control standpoint. Therefore, researchers in this area have often tried to reduce the number of degrees of freedom to as great an extent as is practical (Shields, B.L., Main, J.A., Peterson, S.W., and Strauss, A.M., 1997)."

In designing a prototype hand exoskeleton (Shields, B.L., Main, J.A., Peterson, S.W., and Strauss, A.M., 1997), researchers reduced complexity by reducing DOF to one per finger but discovered problems with this approach. "The human hand has over 25 degrees of freedom, many of which are coupled by the ligamentous structure and location of tendon insertions. This coupling was clear during exoskeleton tests (Shields, B.L., Mian, J. A., Peterson, S.W., and Strauss, A. M., 1997)," in which undesired exoskeleton motion was observed. "One obvious solution to this problem is to add more degrees of freedom to the exoskeleton. This will unfortunately also result in added complexity, weight, and bulk, not to mention a more sophisticated controller (Shields, B.L., Main, J. A., Peterson, S. W., and Strauss, A. M., 1997)."

Researchers involved with the BLEEX lower body exoskeleton took a different approach to this tradeoff. "Each BLEEX leg has 7 DOF..., but actuating all of them creates unnecessarily high- power consumption and control complexity. Instead, only joints that require substantial power should be actuated… [S]ince the primary goal of a lower-extremity exoskeleton is locomotion, the joint power requirements for the BLEEX were determined by analyzing the walking cycle…. (Zoss, A.B., Kazerooni, H., and Chu, A., 2006)" Additionally, the hip and other joints were simplified such that overall the BLEEX represents a "near anthropomorphic" design (Zoss, A. and Kazerooni, H., 2006).

Many present upper body exoskeletons overcome weight or bulk issues by being mounted to a wall or stand (i.e., "grounded"), or to a wheelchair (Lo, H.S. and Xie, S.Q., 2012). This is adequate for applications where a limited and defined workspace is involved, or where a patient requires a wheelchair. While lower body and full body exoskeletons bear their own weight, there are many applications for which a wearable, "ambulatory" orthotic or assistive device is all that is needed. Improvements in mass, power density (see above) and actuation are necessary precursors to widespread use.

The aesthetic appeal of the exoskeleton will eventually have to be addressed, at least for some applications. For example, like many current exoskeletons, WOTAS was designed as a platform to explore a specific concept and not as a final orthotic solution. While it successfully demonstrated the feasibility of mechanical tremor suppression, it was too bulky and heavy to be used day-to-day (Rocon, E. et al., 2007). "The main wish expressed by the potential users was the possibility of hiding the exoskeleton under clothing (Rocon, E. et al., 2007)."

Skin surface EMG signals are often used as a control input because they directly reflect the intentions of the user, but EMG-based control is difficult to realize due to several issues: obtaining the same EMG signals for the same motion is difficult even with the same person, the activity of antagonist muscles affects the joint torque, many muscles are involved in a single joint motion, one muscle is simultaneously involved in more than one motion, the role of each muscle for a certain motion varies in accordance with joint angles, the activity level of some muscles such as bi-articular muscles are affected by the motion of other joints (Kiguchi, K., Rahman, M.H., Sasaki, M., and Teramoto, K., 2008) and the EMG signals can vary due to muscle fatigue (Lalitharatne, T.D., Teramoto, K., Hayashi, Y., Nanayakkara, T., and Kiguchi, K., 2013).

Additional uncertainty is related to the differences between humans and machines. "The exact locations of the human joint axes of rotation cannot be known on living subjects, due to coverage of the joints. Biological joints are not ideal "single DOF" joints, but have rather complex joint surface geometries, which cause shifting axes of rotation during motion. Additionally, fixation of a robotic device on a human limb is never rigid, such that slippage between the device and the limb will occur. This will lead to further misalignment between the mechanism and human joints (Schiele, A. and van der Helm, F.C.T., 2006)," on the order of a few centimeters. Such misalignment can lead to pressure sores on the skin, long-term joint damage, joint dislocation and cartilage damage, and stumbling (Schiele, A. and van der Helm, F.C.T., 2006).

The activity level of each muscle and the way of using each muscle for a certain motion is different between persons (Kiguchi, K., Rahman, M.H, Sasaki, M., and Teramoto, K., 2008). Several solutions proposed to provide adaptive control between users: adjusting impedance (Kiguchi, K. and Hayashi, Y., 2012), myoprocessors with optimization ("gene" modelling) (Cavallaro, E.E., Rosen, J., Perry, J.C., and Burns, S., 2006), adaptive gain (Kang, H. and Wang, J., 2013), neuro fuzzy modifiers (single) (Gopura, R.A.R.C., et al., 2009).

Safety is a paramount concern with robotic systems, especially for robots that must interact with humans. Unfortunately, "there is no industry-standard approach to designing these safety-critical robot systems. Numerous safety-critical software systems have been developed and deployed in other domains ranging from aircraft flight management systems to nuclear power plants (Roderick, S.M. and Carignan, C.R., 2005)." Similar analytical methods, such as fault tree analysis, should be applied to the design of robotic exoskeletons. Some common concerns with these systems are moving the human outside their safe position range, moving the human at an excessive velocity, and applying excessive torque to the human or allowing the human to apply excessive torque against the robot.

The system reaction to fault detection must also be carefully considered. For example, upon fault detection, the system could be commanded to either halt motion or power down the affected motors. Removing power has the undesirable effect of leaving the human to bear the weight of the device, which presents hazards of its own. This approach is only appropriate in response to more severe failures (Roderick, S.M. and Carignan, C.R., 2005).

The safety requirements for mechanical design of the upper body exoskeleton include: “axes deviation of wrist flexion/extension axis and wrist radial/ulnar axis” should be satisfied, “ill effect caused by the movement of the center of rotation of shoulder joint due to upper-arm motions should be canceled out,” and “the mechanical singularity should not be occurred within the workspace of the robot (Gopura, R.A.R.C. and Kiguchi, K., 2009).”

The two main aspects that need good consideration are (Schiele, A., 2007) implementation of the actuation and motor control and intrinsic mechanical and kinematic design of their structure. To ensure the human safety when using exoskeleton, mechanical constraint combined with software limitations are most popular methods. CADEN-7 used mechanical constraints to prevent the excessive movement of body segments. CADEN-7 also used a pulley in design to enable slip when limitation reached. Electrical system of CADEN-7 contained 3 shutoff switches to set electrical constraints. Gupta et al. and Gopura et al. also used mechanical stops and control limitations to ensure safety (Gopura, R.A.R.C. et al., 2009).

REFERENCES

Ali, H. (2014). Bionic Exoskeleton: History, Development and the Future. *International Conference on Advances in Engineering & Technology*.

Army.mil. (n.d.). *Army Explores Futuristic Uniform for SOCOM*. The Official Homepage of the United States Army.

Asbeck, A. T., Dyer, R. J., Larusson, A. F., & Walsh, C. J. (2013). Biologically-Inspired Soft Exosuit. *IEEE International Conference on Rehabilitation Robotics*. 10.1109/ICORR.2013.6650455

Au, A. T. C., Arthur, T. C., & Kirch, R. F. (2000). EMG-Based Prediction of Shoulder and Elbow Kinematics in Able-Bodied and Spinal Cord Injured Individuals. *IEEE Transactions on Rehabilitation Engineering*, *8*(4), 471–480. doi:10.1109/86.895950 PMID:11204038

Beattie, E., McGill, N., Parrotta, N., & Vladimirov, N. (n.d.). *Titan: A Powered, Upper-Body Exoskeleton.* Retrieved from http://www.contact.astm.org/studentmember/images/2012_SParrotta_Paper.pdf

Bouzit, M., Burdea, G., Popescu, G., & Boian, R. (2002, June). The Rutgers Master II - New Design Force-Feedback Glove. *IEEE/ASME Transactions on Mechatronics*, *7*(2), 256–263. doi:10.1109/TMECH.2002.1011262

Brown, M., Tsagarakis, N., & Caldwell, D. G. (2003). Exoskeletons for human force augmentation *Industrial Robot. International Journal (Toronto, Ont.)*, *30*(6), 592–602.

Burgar, C. G., Lum, P. S., Shor, P. C., & Machiel Van der Loos, H. F. (2000). Development of Robots for Rehabilitation Therapy: The Palo Alto VA/Stanford Experience. *Journal of Rehabilitation Research and Development*, *37*, 663–673. PMID:11321002

Casolo, F., Cinquemani, S., & Cocetta, M. (2008). On Active Lower Limb Exoskeletons Actuators. *5th International Symposium on Mechatronics and Its Applications*.

Cavallaro, E. E., Rosen, J., Perry, J. C., & Burns, S. (2006, November). Real-Time Myoprocessors for a Neural Controlled Powered Exoskeleton Arm. *IEEE Transactions on Biomedical Engineering*, *53*(11), 2387–2396. doi:10.1109/TBME.2006.880883 PMID:17073345

Chu, A., Kazerooni, H., & Zoss, A. (2005). On the Biomimetic Design of the Berkeley Lower Extremity Exoskeleton (BLEEX). *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 4345-4352. 10.1109/ROBOT.2005.1570789

Daerden, F., & Lefeber, D. (2002). Pneumatic Artificial Muscles: Actuators for Robotics and Automation. *European Journal of Mechanical and Environmental Engineering*, *47*(1), 11–21.

Dellon, B., & Matsuoka, Y. (2007, March). Prosthetics, Exoskeletons, and Rehabilitation: Now and for the Future. *IEEE Robotics & Automation Magazine*. doi:10.1109/MRA.2007.339622

Dollar, M. A., & Herr, H. (2008, February). Lower Extremity Exoskeletons and Active Orthoses: Challenges and State of the Art. *IEEE Transactions on Robotics*, *24*(1), 144–158. doi:10.1109/TRO.2008.915453

Dombrowksi, P., & Coval, D. (2012). Private First-Class Wing Man. *J Swanson Sch Eng, 4*.

Ferris, D. P., & Lewis, C. L. (2009). Robotic Lower Limb Exoskeletons Using Proportional Myoelectric Control. *Conference Proceedings; ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*. PMID:19964579

Foundation for Spinal Cord Injury Prevention Care and Cure. (n.d.). Retrieved from http://www.fscip.org/facts.htm

Garrec, P., Friconneau, J. P., Measson, Y., & Perrot, Y. (2008). ABLE, An Innovative Transparent Exoskeleton for the Upper-Limb. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1483-1488). IEEE. 10.1109/IROS.2008.4651012

Gopura, R. A. R. C., & Kiguchi, K. (2009). Mechanical designs of Active Upper-Limb Exoskeleton Robots: State-of-the-art and Design Difficulties. *IEEE International Conference on Rehabilitation Robotics (ICORR)*, 178-187. 10.1109/ICORR.2009.5209630

Gopura, R. A. R. C., & Kiguchi, K. (2009). SUEFUL-7: A 7DOF Upper-Limb Exoskeleton Robot with Muscle-Model-Oriented EMG- Based Control. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 1126-1131. 10.1109/IROS.2009.5353935

Guizzo, E., & Goldstein, H. (2005). The Rise of the Body Bots [Robotic Exoskeletons]. Spectrum, IEEE, 42(10), 50-56.

Harwin, W., Leiber, L., Austwick, G., & Dislis, C. (1998). Clinical Potential and Design of Programmable Mechanical Impedances for Orthotic Applications. *Robotica*, *16*(5), 523–530. doi:10.1017/S026357479800068X

Herr, H. (2009). Exoskeletons and Orthoses: Classification, Design Challenges, and Future Directions. *Journal of Neuroengineering and Rehabilitation*, *6*(21). PMID:19538735

Hogan, N., Krebs, H.I., Sharon, A., & Charnarong, J. (1995). *Interactive Robot Therapist*. Academic Press.

Housman, S. J., Le, V., Rahman, T., Sanchez, R. J., & Reinkensmeyer, D. J. (2007). Arm-Training with T-WREX After Chronic Stroke: Preliminary Results of a Randomized Controlled Trial. In *ICORR 2007. IEEE 10th International Conference on Rehabilitation Robotics* (pp. 562-568). IEEE. 10.1109/ICORR.2007.4428481

HULC®. (n.d.). Lockheed Martin.

Ito, K., Tsuji, T., Kato, A., & Ito, M. (1992). EMG Pattern Classification for a Prosthetic Forearm with Three Degrees of Freedom. *IEEE International Workshop on Robot and Human Communication*, 69-74. 10.1109/ROMAN.1992.253907

Kang, H., & Wang, J. (2013, July). Adaptive Control of 5 DOF Upper-Limb Exoskeleton Robot with Improved Safety. *ISA Transactions*, *52*(6), 844–852. doi:10.1016/j.isatra.2013.05.003 PMID:23906739

Kazerooni, H. (2005). *Exoskeletons for Human Power Augmentation*. IEEE. doi:10.1109/IROS.2005.1545451

Kazerooni, H., Steger, R., & Huang, L. (2006). Hybrid Control of the Berkeley Lower Extremity Exoskeleton (BLEEX). *The International Journal of Robotics Research*, 561-573.

Kiguchi, K., & Hayashi, Y. (2012, August). An EMG-Based Control for an Upper-Limb Power-Assist Exoskeleton Robot. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, *42*(4), 1064–1071. doi:10.1109/TSMCB.2012.2185843 PMID:22334026

Kiguchi, K., Rahman, M. H., Sasaki, M., & Teramoto, K. (2008). Development of a 3DOF mobile exoskeleton robot for human upper-limb motion assist. *Robotics and Autonomous Systems*, *56*(8), 678–691. doi:10.1016/j.robot.2007.11.007

Kirsh, R. F., & Au, A. T. C. (1997). EMG-Based Motion Intention Detection for Control of a Shoulder Neuroprosthesis. *IEEE International Conference of the Engineering in Medicine and Biology Society*, 5, 1944-1945. 10.1109/IEMBS.1997.758719

Krebs, H. I., Hogan, N., Aisen, M. L., Hogan, N., McDowell, F., & Volpe, B. T. (1997). The Effect of Robot-Assisted Therapy and Rehabilitative Training on Motor Recovery Following Stroke. [PubMed]. *Archives of Neurology*, *54*(4), 443–446. doi:10.1001/archneur.1997.00550160075019

Kuribayashi, K., Shimizu, S., Okimura, K., & Taniguchi, T. (1993). A Discrimination System Using Neural Networks for EMG-Control Prostheses-Integral type of EMG Signal Processing. *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1750-1755.

Lalami, M. (2013). Output Feedback Control of an Actuated Lower Limb Orthosis with Bounded Input. *International Journal of Industrial Robots*, *40*(6).

Lalitharatne, T. D., Teramoto, K., Hayashi, Y., Nanayakkara, T., & Kiguchi, K. (2013). Evaluation of Fuzzy-Neuro Modifiers for Compensation of the Effects of Muscle Fatigue on EMG-Based Control to be Used in Upper-Limb Power-Assist Exoskeletons. *Journal of Advanced Mechanical Design, Systems and Manufacturing*, *7*(4), 736–751. doi:10.1299/jamdsm.7.736

Lo, H. S., & Xie, S. Q. (2012). Exoskeleton robots for upper-limb rehabilitation: State of the art and future prospects. *Medical Engineering & Physics*, *34*(3), 261–268. doi:10.1016/j.medengphy.2011.10.004 PMID:22051085

Lucas, L., DiCicco, M., & Matsuoka, Y. (2004). An EMG-Controlled Hand Exoskeleton for Natural Pinching. *Journal of Robotics and Mechatronics*, *16*(5), 482–488. doi:10.20965/jrm.2004.p0482

Lum, P. S., Burgar, C. G., Van der Loos, M., Shor, P. C., Majmundar, M., & Yap, R. (2006). MIME Robotic Device for Upper-Limb Neurorehabilitation in Subacute stroke subjects: A follow-up study. *Journal of Rehabilitation Research and Development*, *43*(5), 631. doi:10.1682/JRRD.2005.02.0044 PMID:17123204

Mao, Y., & Agrawal, S. K. (2012). Design of a Cable-Driven Arm Exoskeleton (CAREX) for Neural Rehabilitation. *IEEE Transactions on Robotics*, *28*(4), 922–931. doi:10.1109/TRO.2012.2189496

Mihelj, M., Nef, T., & Riener, R. (2007). ARMin II-7 DoF Rehabilitation Robot: Mechanics and Kinematics. In *IEEE International Conference on Robotics and Automation* (pp. 4120-4125). IEEE. 10.1109/ROBOT.2007.364112

Miller, C., Ruiz-Torres, R., Stienen, A. H. A., & Dewald, J. P. A. (2009). A Wrist and Finger Force Sensor Module for Use During Movements of the Upper Limb in Chronic Hemiparetic Stroke. *IEEE Transactions on Biomedical Engineering*, *56*(9), 2312–2317. doi:10.1109/TBME.2009.2026057 PMID:19567336

Mizen, N. J. (1965). *Preliminary Design for the Shoulders and Arms of a Powered, Exoskeletal Structure.* Cornell Aeronautical Laboratory Report VO-1692-V-4.

Nef, T., Mihelj, M., & Riener, R. (2007). ARMin: A Robot for Patient-Cooperative Arm Therapy. *Medical & Biological Engineering & Computing*, *45*(9), 887–900. doi:10.100711517-007-0226-6 PMID:17674069

Perry, J., & Rosen, J. (2006). Design of a 7 degree-of-freedom upper-limb powered exoskeleton. *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, 805 –810. 10.1109/BIOROB.2006.1639189

Perry, J. C., Rosen, J., & Burns, S. (2007). Upper-limb powered exoskeleton design. *Mechatronics, IEEE/ASME Transactions on, 12*(4), 408-417.

Pneumatic-Hydraulic Actuator System. (n.d.). *Patent US 4647004A*. Washington, DC: US Patent Office.

Pons, J. L., Moreno, J. C., Brunetti, F. J., & Rocon, E. (2007). Lower-limb Wearable Exoskeleton. In *Rehabilitation Robotics* (pp. 772-474). InTech. Available from: http://www.intechopen.com/books/rehabilitation_robotics/lower-limb_wearable_exoskeelton, 2007.

Pons, J.L., Rocon, E., Ruiz, A.F., & Morenso, J.C. (2007). Upper-Limb Robotic Rehabilitation Exoskeleton: Tremor Suppression. *Rehabilitation Robotics*, 453-470.

Pratt, J. E., Krupp, B. T., & Morse, C. J. (2014). The RoboKnee: An Exoskeleton for Enhancing Strength and Endurance During Walking. *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*.

Raytheon, X. O. S. 2 Exoskeleton, Second-Generation Robotics Suit, United States of America. (n.d.). Retrieved from http://www.army-technology.com/projects/raytheon-xos-2-exoskeleton-us/

Reinkensmeyer, D. J., Dewald, J. P. A., & Rymer, W. Z. (1999). Guidance-Based Quantification of Arm Impairment Following Brain Injury: A pilot study. *IEEE Transactions on Rehabilitation Engineering*, *7*(1), 1–11. doi:10.1109/86.750543 PMID:10188602

Ren, Y., Park, H., & Zhang, L. (2009). Developing a Whole-Arm Exoskeleton Robot with Hand Opening and Closing Mechanism for Upper Limb Stroke Rehabilitation. *IEEE 11th International Conference on Rehabilitation Robotics*.

ReWalk Robotics Announces Reimbursement Coverage by Major German Insurance Company. (n.d.). Retrieved from http://www.rewalk.com/rewalk-robotics-announces-reimbursement-coverage-by-major-german- insurance-company/

Rocon, E., Belda-Lois, J. M., Ruiz, A. F., Manto, M., Moreno, J. C., & Pons, J. L. (2007, September). Design and Validation of a Rehabilitation Robotic Exoskeleton for Tremor Assessment and Suppression. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *15*(3), 367–378. doi:10.1109/TNSRE.2007.903917 PMID:17894269

Roderick, S. N., & Carignan, C. R. (2005). An Approach to Designing Software Safety Systems for Rehabilitation Robots. In *Proceedings of the 2005 IEEE 9th International Conference on Rehabilitation Robotics* (pp. 252-257). IEEE. 10.1109/ICORR.2005.1501096

Rosen, J., Perry, J. C., Manning, N., Burns, S., & Hannaford, B. (2005). The Human Arm Kinematics and Dynamics During Daily Activities-Toward a 7 DOF Upper Limb Powered Exoskeleton. In *ICAR'05. Proceedings of 12th International Conference on Advanced Robotics* (pp. 532-539). IEEE. 10.1109/ICAR.2005.1507460

Sanchez, R. J., Liu, J., Rao, S., Shah, P., Smith, R., Rahman, T., & Reinkensmeyer, D. J. (2006). Automating Arm Movement Training Following Severe Stroke: Functional Exercises with Quantitative Feedback in a Gravity-Reduced Environment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *14*(3), 378–389. doi:10.1109/TNSRE.2006.881553 PMID:17009498

Sansoni, S., Wodehouse, A., & Buis, A. (2014). *The Aesthetics of Prosthetic Design: From Theory to Practice*. International Design Conference – Design, Dubrovnik, Croatia.

Sarakoglou, I., Tsagarakis, N. G., & Caldwell, D. G. (2004). Occupational and Physical Therapy Using a Hand Exoskeleton Based Exerciser. *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Sawicki, G. S., & Ferris, D. P. (2009, June). A Pneumatically Powered Knee-Ankle Foot Orthosis (KAFO) with Myoelectric Activation and Inhibition. *Journal of Neuroengineering and Rehabilitation*, *6*(1), 23. doi:10.1186/1743-0003-6-23 PMID:19549338

Schabowksy, C.N., Godfrey, S.B., Holley, R.J., & Lum, P.S. (2010). Development and Pilot Testing of HEXORR: Hand EXOskeleton Rehabilitation Robot. *Journal of NeuroEngineering and Rehabilitation, 7*(37).

Schiele, A. (2007). Undesired Constraint Forces in Non-Ergonomic Wearable Exoskeletons. *Extended Abstract for IROS'07 Workshop on Assistive Technologies: Rehabilitation and Assistive Robotics*.

Schiele, A., & van der Helm, F. C. T. (2006, December). Kinematic Design to Improve Ergonomics in Human Machine Interaction. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *14*(4), 456–469. doi:10.1109/TNSRE.2006.881565 PMID:17190037

Schnieders, T. M., Stone, R. T., Oviatt, T., & Danford-Klein, E. (2017). ARCTiC LawE- An Upper Body Exoskeleton for Firearm Training. *Augmented Human Research*, *2*(1), 1. doi:10.100741133-017-0004-4

Shields, B. L., Main, J. A., Peterson, S. W., & Strauss, A. M. (1997, September). An Anthropomorphic Hand Exoskeleton to Prevent Astronaut Hand Fatigue During Extravehicular Activities. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans*, *27*(5), 668–673. doi:10.1109/3468.618265 PMID:11541130

Singh, R. M., & Chatterji, S. (2012). Trends and Challenges in EMG Based Control Scheme of Exoskeleton Robots – A Review. *International Journal of Scientific & Engineering Research*, *3*(8).

Soft Exosuit. Lightweight suit to increase the wearer's strength and endurance. (n.d.). Wyss Institute. Retrieved from http://wyss.harvard.edu/viewpage/456

Song, Z., & Guo, S. (2011). Design Process of Exoskeleton Rehabilitation Device and Implementation of Bilateral Upper Limb Motor Movement. *Journal of Medical and Biological Engineering*, *32*(5), 323–330. doi:10.5405/jmbe.987

Stergiopoulos, P., Fuchs, P., & Laurgeau, C. (2003). *Design of a 2-Finger Hand Exoskeleton for VR Grasping Simulation*. EuropHaptics.

Tzafestas, C. S. (2003). Whole-Hand Kinesthetic Feedback and Haptic Perception in Dexterous Virtual Manipulation. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans*, *33*(1), 100–113. doi:10.1109/TSMCA.2003.812600

Uemura, M., Kanaoka, K., & Kawamura, K. (2006). Power Assist Systems Based on Resonance of Passive Elements. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4316-4321. 10.1109/IROS.2006.281965

Umetani, Y., & Yamada, Y. (1999). *Skil Mate. In Wearable Exoskeleton Robot* (pp. 984–988). IEEE.

US Army to Build Armored Talos Suit That Merges Man and Machine. (n.d.). Singularity HUB.

Vukobratovic, M., Ciric, V., & Hristic, D. (1972). Contribution to the Study of Active Exoskeletons. *Proceedings of the 5th International Federation of Automatic Control Congress*.

Winder, S. B., & Esposito, J. M. (2008, March). Modeling and control of an upper-body exoskeleton. In *System Theory, 2008. SSST 2008. 40th Southeastern Symposium on* (pp. 263-268). IEEE. 10.1109/SSST.2008.4480234

Yuan, P., Wang, T., Ma, F., & Gong, M. (2014). Key Technologies and Prospects of Individual Combat Exoskeleton. Knowledge Engineering and Management, 305-316. doi:10.1007/978-3-642-37832-4_28

Zeilig, G., Weingarden, H., Zwecker, M., Dudkiewicz, I., Bloch, A., & Esquenazi, A. (2012). Safety and Tolerance of the ReWalk™ Exoskeleton Suit for Ambulation by People with Complete Spinal Cord Injury: A Pilot Study. *The Journal of Spinal Cord Medicine*, *35*(2), 101–196. doi:10.1179/2045772312Y.0000000003 PMID:22333043

Zoss, A., & Kazerooni, H. (2006). Design of an Electrically Actuated Lower Extremity Exoskeleton. *Advanced Robotics*, *20*(9), 967–988. doi:10.1163/156855306778394030

Zoss, A. B., Kazerooni, H., & Chu, A. (2006, April). Biomechanical Design of the Berkeley Lower Extremity Exoskeleton (BLEEX). *IEEE/ASME Transactions on Mechatronics*, *11*(2), 128–138. doi:10.1109/TMECH.2006.871087

Chapter 9
Awareness-Based Recommendation by Passively Interactive Learning:
Toward a Probabilistic Event

Tomohiro Yamaguchi
National Institute of Technology, Nara College, Japan

Takuma Nishimura
Nippon Telegraph and Telephone West Corporation, Japan

Shota Nagahama
National Institute of Technology, Nara College, Japan

Keiki Takadama
The University of Electro-Communications, Japan

ABSTRACT

In artificial intelligence and robotics, one of the important issues is to design human interface. There are two issues: One is the machine-centered interaction design. Another one is the human-centered interaction design. This research aims at the latter issue. This chapter presents the interactive learning system to assist positive change in the preference of a human toward the true preference. Then evaluation of the awareness effect is discussed. The system behaves passively to reflect the human intelligence by visualizing the traces of his/her behaviors. Experimental results showed that subjects are divided into two groups, heavy users and light users, and that there are different effects between them under the same visualizing condition. They also showed that the authors' system improves the efficiency for deciding the most preferred plan for both heavy users and light users. As future research directions, a probabilistic event and its basic recommendation way are discussed.

DOI: 10.4018/978-1-5225-5276-5.ch009

INTRODUCTION

Interactive Reinforcement Learning with Human

A long term goal of interactive learning system is to incorporate human to solve complex tasks. Reinforcement learning is the Standard behavior learning method for among robot, animal and human. In interactive reinforcement learning, there are two roles, a learner and a trainer. The input of a reinforcement learner as a learning goal is called a *reward*, and the output of the learner as a learning result is called a *policy*. For example, as training a dog by a human trainer, Peterson (2000, 2001) showed that clicker training is an easy way to shape new behaviors. When a dog performs a new behavior to learn, the trainer clicks the clicker as a positive reward. Pryor (2006) remarks that clicker training is a method for training an animal that uses positive reinforcement in conjunction with a clicker to mark the behavior being reinforced under behavior modification principles.

In current researches of interactive reinforcement learning, there are two approaches to support a learner by giving feedback as, whether a learning goal (reward based), or a learning result (policy based). The former approach is clicker training for the robot, in that a human trainer gives a learning goal to the robot learner. In field of robot learning, Kaplan et al. (2002) showed that interactive reinforcement learning method in that reward function denoting goal is given interactively has worked to establish the communication between a human and the pet robot AIBO. The main feature of this method is the interactive reward function setup which was fixed and build-in function in the main feature of previous reinforcement learning methods. So the user can sophisticate reinforcement learner's behavior sequences incrementally.

Ng et al. (1999) and Konidaris & Barto (2006) showed that reward shaping is the theoretical framework of such interactive reinforcement learning methods. Shaping is to accelerate the learning of complex behavior sequences. It guides learning to the main goal by adding shaping reward functions as subgoals. Previous reward shaping methods have three assumptions on reward functions as following:

- Main goal is given or known for the designer;
- Marthi (2007) remarks that subgoals are assumed as shaping rewards those are generated by potential function to the main goal;
- Ng et al. (1999) showed that shaping rewards are policy invariant, it means not affecting the optimal policy of the main goal.

However, these assumptions will not be true on interactive reinforcement learning with a non-expert end-user. Main reason is discussed by Griffith et al. (2013) that human feedback signals may be inconsistent with the optimal policy. It is not easy to keep these assumptions while the end-user gives rewards for the reinforcement learning agent. It is that the reward function may not be fixed for the learner if an end-user changes his/her mind or his/her preference. However, most of previous reinforcement learning methods assume that the reward function is fixed and the optimal solution is unique, so they will be useless in interactive reinforcement learning with an end-user.

To avoid this problem, the latter approaches are that a human trainer provides a sample of learning result to the robot learner. For robot learning with human, inverse reinforcement learning proposed by Ng & Russell (2000) is the method that after the human provides demonstrations of an optimal policy, the reward function for the demonstrations is generated to learn the optimal policy. Another approach is called policy shaping proposed by Griffith et al. (2013). Instead of requiring demonstrations, it allows a human trainer to simply critique the learner's behavior ("that was right/wrong"). Thus the human's feedback is a label on the optimality of actions of each state.

To introduce our approach, we organize reinforcement learning methods. Table 1 shows the characteristics on interactive reinforcement learning. In reinforcement learning, an optimal solution is decided by the reward function and the optimality criteria. In standard reinforcement learning, an optimal solution is fixed since both the reward function and the optimality criteria are fixed. On the other hand, in interactive reinforcement learning, an optimal solution may change according to the interactive reward function. Furthermore, in interactive reinforcement learning with human, various optimal solutions will occur since the optimality criteria depend on human's preference.

Then the objective of this research is to recommend preferable solutions of each user. The main problem is "how to guide to estimate the user's preference?" Our solution consists of two ideas. One is to prepare various solutions by *every-visit-optimality* proposed by Satoh & Yamaguchi (2006), another is the *coarse to fine recommendation* strategy proposed by Yamaguchi, Nishimura & Sato (2011). Our approach considers a human as a novice trainer. First, the novice trainer inputs initial

Table 1. Characteristics on interactive reinforcement learning

Type of Reinforcement Learning	An Optimal Solution	Reward Function	Optimality Criteria
standard	fixed	fixed	fixed
interactive	may change	interactive	fixed
interactive with human	various optimal	may change	human's preference

learning goals, then the learning system generates and suggests the candidates of the optimal leaning result to the novice trainer in order to make clear his/her final learning goals.

Requirements for Interactive System as Being Human Adaptive and Friendly

Recent years there are many researches on recommender systems to present information items that are likely of interest to the user. A recommender system is one of the major intelligent & interactive systems with a human user. However, it sometimes seems to be officious for the user. Yamaguchi et al. (2012) discussed that main problem is that it is active but less intelligent.

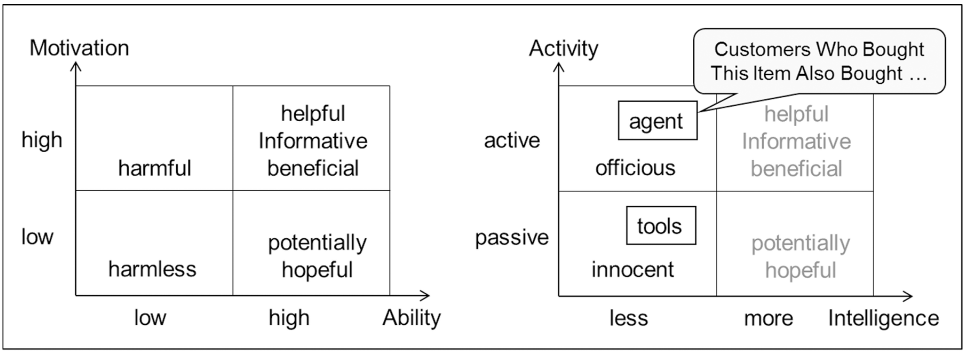
For examples, Office Assistant (Office97-2003) and the recommender system by Amazon. com are typical intelligent & interactive systems. Office Assistant is to assist users by offering advice based on Bayesian algorithms. However, the feature evoked a strong negative response from many users. On the other hand, the recommender system by Amazon.com is based on collaborative filtering proposed by Schafer et al. (2001). After a user views some items in the amazon's web site, the recommender system shows information to the user such as "What Do Customers Ultimately Buy After Viewing This Item?" or "Customers Who Bought This Item Also Bought ..." However, it sometimes seems to be officious. Why? The key point is that these intelligent & interactive systems are active but less intelligent.

Useful suggestion is from social psychology in 1970's. In the *human performance theory* Anderson & Butzin (1974) argued that a human's performance depends on both his ability and his motivation. Figure 1 shows the human performance theory applied to an agent. Figure 1(a) shows a human's performance relation. The better cases are when the ability is high. It is beneficial if high motivation, or it is potentially hopeful if low motivation. On the other hand, if the ability is low, the bad case is not low motivation (harmless) but high motivation (harmful). Therefore, the worst case is the combination of low ability and high motivation.

Then we define that an agent's performance depends on both its intelligence and its activity. Figure 1(b) shows an agent's performance relation. Note that the agent's intelligence axis is the ability axis in figure 1(a), and that the agent's activity axis is the motivation axis in figure 1(a). Such as the human's performance relation in figure 1(a), the worst case in figure 1(b) is the combination of low intelligence and high activity. This is the reason why active but less intelligent agents seem to be officious.

To solve this problem, less active but more intelligent agent is desirable. Therefore, objective of the research is to realize harmless or potentially desirable interactive

Figure 1. The human performance theory applied to an agent



(a) Human's performance relation *(b) Agent's performance relation*

recommendation for a human user by our less active recommendation agent. There are several merits. First, it allows a user's preference shift since it performs less active recommendation. Second, it can assist the user for its convergence of his preference. Therefore, we aim to design to assist the awareness for a user.

How to Keep the Current Preference of a User?

Another problem is to keep the current preference of a user. Typically, a recommender system compares the user's profile to some reference characteristics, and seeks to predict the 'rating' that a user would give to an item they had not yet considered. These characteristics may be from the information item (the content-based approach) or the user's social environment (the collaborative filtering approach) researched by Balabanovic & Shoham (1997), Riecken (2000), and Schafer et al. (2001).

One of the main research issues is building the user's profile. There are two ways whether explicit data collection or implicit data collection. The explicit data collection is asking a user that he/she likes, or to choose the best one. The implicit data collection is observing behaviors of a user, for example the record of items that the user views or chooses. However, there is a problem that the collected user's profile is not same as the user's current preference, we called it *the preference change problem* proposed by Yamaguchi et al. (2009). In other words, these systems often recommend irerelevant items for the user since most previous researches do not directly specify the user's current preference. Consequently, it is necessary to treat the ambiguity, shift, or changes of a user's preference.

Passive Interaction Design to Assist the Awareness for a Human

To solve these problems, first we define the model of a user's *preference shift* for the recommendation space by two axes, preference *reduction* and preference *extension*. Then the system prepares various recommendation plans according to the goals that the user selected by our extended reinforcement learning method proposed by Konda et al. (2002a, 2002b), to assist the user's awareness for his/her *preference shift*, we propose the user-centered recommendation by visualizing both the recommendation space with prepared recommendation plans and the user's preference trace as the history of the recommendation in it. Note that the recommendation space visualizes the possible *preference shift* of the user. We consider the interaction between a user and a recommender system to be the cooperative learning in human agent interaction. We assume that the process to make clear the user's preference is to *be aware the true preference* of himself. To realize this process, Yamaguchi et al. (2009) and Takadama et al. (2012) discuss that it is necessary to support the user's awareness. Note that supporting the awareness means not to prevent it but to assist it. The experimental results show that our system improves the efficiency for deciding the most preferred plan.

BACKGROUND

This section describes an overview of our plan recommendation system. First, we introduce the plan recommendation task and the model of a user's *preference shift*. Then our plan recommendation system based on our extended reinforcement learning called LC-learning is described. As for details, please refer Satoh & Yamaguchi (2006) or Yamaguchi et al. (2011).

The Round-Trip Plan Recommendation Task

The task for a user is to decide the most preferred round-trip plan after selecting four cities to visit among eighteen cities. The task for the system is to estimate the preferable round-trip plans to each user and to recommend them sequentially. The way of generating these plans by LC-learning will be described in subsection "Overview of the interactive plan recommendation system".

The plan recommendation procedure is as follows:

- **Step1:** A user selects four cities to visit. These are called goals.
- **Step2:** Various round-trip plans including at least one goal are recommended.
- **Step3:** The user decides the most preferred round-trip plan among them.

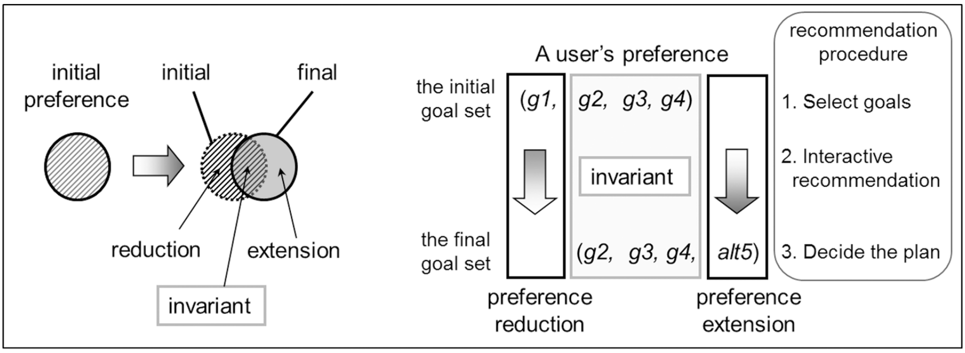
The Model of a User's Preference Shift

Figure 2 shows the model of a user's preference shift. It is defined by two-axes space. Figure 2(a) illustrates the model. Comparing the previous preference set and the current preference set, the common set is the invariant preference, the reduction set is called *preference reduction*, and the addition set is called *preference extension*.

Next we explain the significance of our model on user profiling. There are two ways whether the data collection is explicit or implicit. From the viewpoint of the explicit data collection, our mode aims for realizing not similar but various recommendations with a set of goal cities by a user. From the viewpoint of the implicit data collection, our model makes the user aware of his preference shift by visualizing the user's preference trace.

Now we describe an illustrated example of the model of a user's preference shift as shown in figure 2(a) according to the plan recommendation procedure in previous subsection. Figure 2(b) shows the illustrated example of the model of a user's preference shift. To begin with, a user selects four goals (g1, g2, g3, g4) as the initial goal set in the first step as shown in previous subsection. After the second step of interactive recommendation, the user decides the most preferred plan including the final goal set as (g2, g3, g4, alt5) in the third step. By comparing the final goal set with the initial goal set, the invariant preference is (g2, g3, g4), the preference reduction is (g1), and the preference extension is (alt5).

Figure 2. The model of a user's preference shift



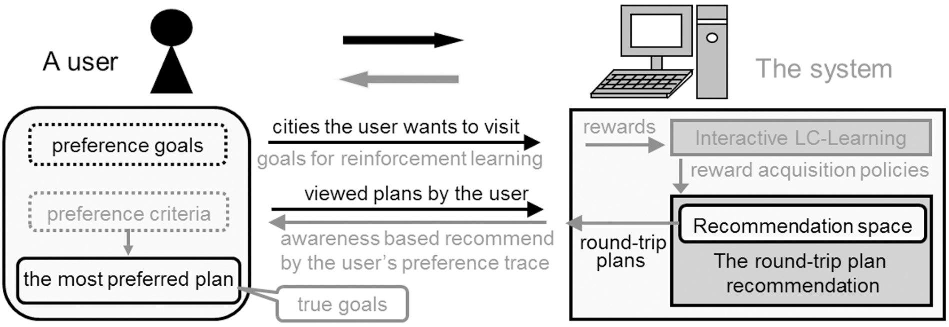
(a) The model *(b) An illustrated example of the model*

Overview of the Interactive Plan Recommendation System

Figure 3 shows the overview of the interactive plan recommendation system. When a user input several goals to visit constantly as his/her preference goals, they are converted to the set of rewards in the plan recommendation block for the input of interactive LC-learning block proposed by Satoh & Yamaguchi (2006). Note that a goal is a reward to be acquired, and a plan means a cycle that acquires at least one reward in a policy. Interactive LC-Learning block is based on extended model-based Average Reward reinforcement learning proposed by Mahadevan (1996). In that, the simple MDP model described by Puterman (1994) is used. Our learning agent consists of three blocks those are model identification block, optimality criterion block and policy search block. The novelty of our method lies in optimality criterion as every-visit-optimality and the method of policy search collecting various reward acquisition policies discussed by Yamaguchi et at. (2011).

After various reward acquisition policies are collected, each policy is output as a round plan that includes at least one preference goal of the user for recommendation to the user. The system displays the recommendation space in that these round plans are located by the preference reduction axis and the preference extension axis. In the recommendation space, the user can view the plans, then the trace of viewed plans is recorded to display as the preference trace in the recommendation space. The preference trace displays the reference time of each plan as the density of the color of each plan icon in the recommendation space. Therefore, the user comes into focus on his/her preference criteria through the interactive recommendation process. The interactive recommendation will finish after the user decides the most preferred plan, then the user finds his/her true goals.

Figure 3. The overview of the interactive plan recommendation system



Overview of the Passive Interactive Recommendation

This subsection describes the interactive recommendation space and the passive recommendation strategy by our system. In the recommendation space, the user can view and select various plans actively. The recommendation space consists of two dimensions, the preference reduction axis and the preference extension axis, in that, various plans are arranged in a plane. As the passive recommendation strategy, we introduce coarse to fine recommendation strategy proposed by Yamaguchi, Nishimura & Sato (2011) that consists of two steps, coarse recommendation step along the preference reduction axis and fine recommendation step along the preference extension axis. Figure 4 shows coarse to fine recommendation strategy in the recommendation space.

1. Grouping various plans by the visited goals.

After preparing various round plans, they are merged into group at the round-trip plan recommendation block. Figure 4(a) shows grouping various plans by the number of included goals. When three goals g1, g2, and g3 are input by a user, then, Group1 in figure 3(a) holds various plans including all goals, g1, g2, and g3. Group2 holds various plans including all but one among g1, g2, or g3, and Group3 holds various plans including all but two among g1, g2, or g3.

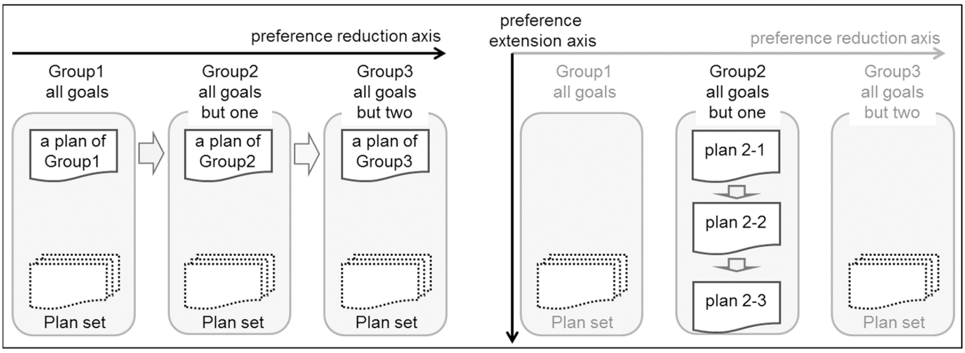
2. Coarse recommendation step.

For the user, the aim of this step is to select a preferable group according to the preference reduction axis. To support the user's decision, the system passively recommends a representative plan in each selected group to the user. Figure 4(a) shows a coarse recommendation sequence when a user changes his preferable group as Group1, Group2, and Group3 sequentially. When the user selects a group, the system presents a representative plan in the group as recommended plan.

3. Fine recommendation step.

For the user, the aim of this step is to decide the most preferable plan in the selected group in previous step according to the preference extension axis. To support the user's decision, the system passively recommends plans among his/her selected group to the user. Figure 4(b) shows a fine recommendation sequence after the user selects his/her preferable group as Group2. In each group, plans are ordered according to the length of a plan.

Figure 4. Coarse to fine recommendation strategy in the recommendation space



(a) Coarse recommendation step

(b) Fine recommendation step

AWARENESS BASED RECOMMENDATION

To assist the user's awareness for his/her preference shift, we propose the user-centered recommendation by visualizing both the recommendation space with prepared recommendation plans and the user's preference trace as the history of the recommendation in it. The recommendation space visualizes the possible preference shift of the user.

Overview of the GUI

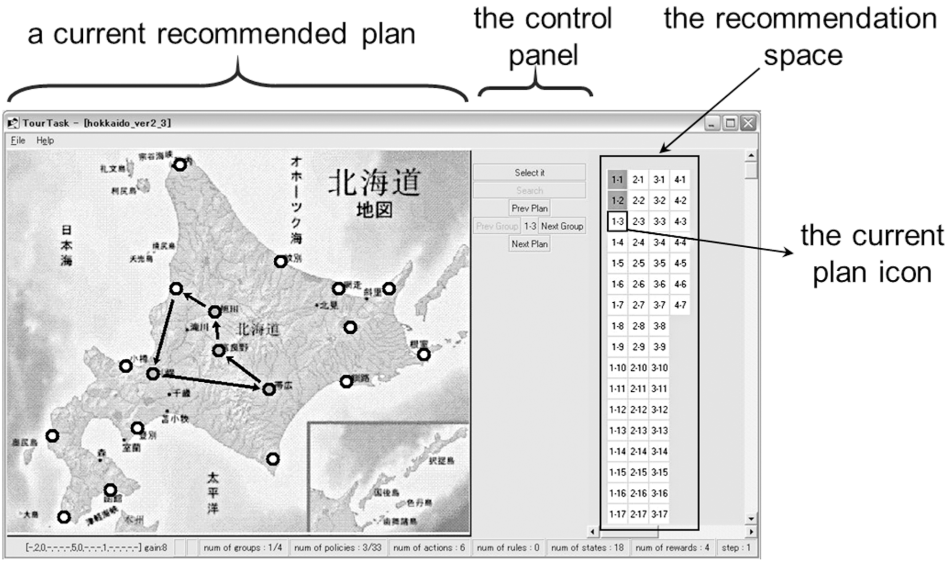
Figure 5 shows the overview of the GUI of our interactive recommendation system. A current recommendation plan in the Hokkaido map is displayed in the left side. In the right mid area, there is the control panel in which a current recommendation plan can move up, down, right or left in the recommendation space. The recommendation space is displayed in the rightmost. Each plan icon is arranged in a plane in the recommendation space. The boxed plan icon in it indicates the current recommendation plan.

Visualizing the Recommendation Space

This subsection describes an overview of the visualization of the *recommendation space*. The objective of this visualization is to inform a user of two kinds of information. First is that the recommendation space consists of two-axes. Second is that in each axis, groups or plans are ordered according to the recommendation order.

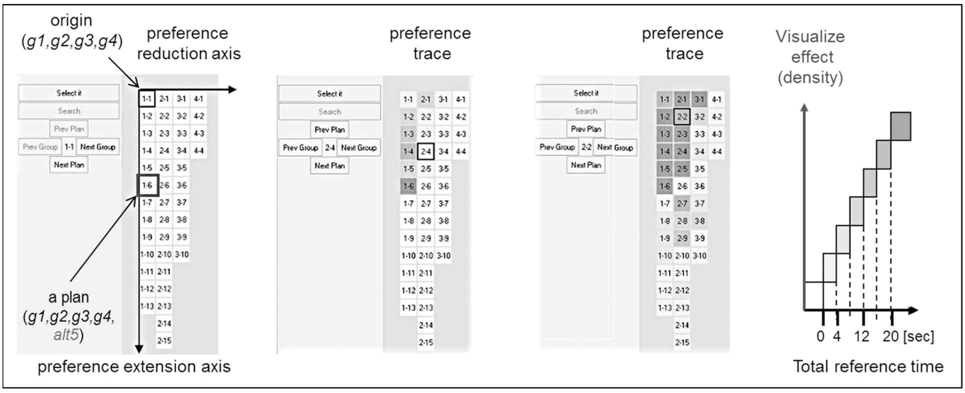
Figure 6 shows a flow of the preference trace in the recommendation space for four goal cities. Figure 6(a) shows an example of the beginning of the recommendation

Figure 5. The overview of the GUI of our interactive recommendation system



[-2,0,-,-,-,-,5,0,-,-,-,1,-,-,-,-,-] gain:8 | num of groups : 1/4 | num of policies : 3/33 | num of actions : 6 | num of rules : 0 | num of states : 18 | num of rewards : 4 | step : 1

Figure 6. A flow of the preference trace in the recommendation space for four goal cities



(a) The beginning (b) The middle (c) The end (d) visualizing rule

space of a user. Each plan icon in it has the plan-id x-y consists of group-id x and plan-number y. In figure 6(a), after a user inputs four cities as goals, these are highlighted by red color. The plan 1-1 is origin of the recommendation space that is the minimum length plan visiting all four goals. The recommendation space is defined by two-axes as follows:

- Coarse axis (preference reduction);
- Fine axis (preference extension).

The horizontal axis is coarse axis that means the group-id for plans defined by Equation (1). group-id is the horizontal distance between plan 1-1 at the origin and the group of the current plan. Note that N0 is the number of initial goals (default is four), and Ni is the number of included goals in each recommended plan. A larger group-id is less familiar but fresh to the user since the plans in it include alternative cities without initial goal set of the user:

$$\text{group-id} = N_0 - N_i + 1 \tag{1}$$

The vertical axis is fine axis that means the plan-number of each group. The order is according to the length of the plans from shorter to longer. Each plan-number y is the vertical distance between the current plan x-y of the group x and uppermost plan x-1, it is about the number of alternative cities in the current plan.

Visualizing the User's Preference Trace

This subsection illustrates the visualizing process of the user's preference trace. The objective of this visualization is to show the distribution and the degree of the user's preference to him/herself. Figure 6(b) shows the middle of the preference trace of the user. The preference trace is visualized as the set of density of the background color of each plan icon. It indicates the total reference time of each plan by the user. In figure 5(b), the preference trace is mainly distributed in a vertical direction, so that the user referred six plans in the Group1 those are plans with four goals. Note that the current plan 2-4 is one of the plans with three goals within four goals. Within plan 1-1 to 1-6, plan 1-6 shows the highest density of the color so that the user watched this plan with the longest time.

Figure 6(c) shows the end of the preference trace of the user. The preference trace shows that the user has referred all recommended plans from plan 1-1 to 4-4. Then the distribution of the density of background color indicates the bias of the user's preference trace. By these visualizations, the user can notice the distribution or the bias of the reference time of each plan in the recommendation space. Figure 6(d) shows the visualizing rule of the preference trace.

Effect of Visualizing the Degree of Preference

Next we analyze the meaning of visualizing the reference time of each plan. If we assume that the reference time corresponds to the degree of preference of the user, the user can notice the distribution or the bias of the preference of each plan in the recommendation space as follows:

- The plan with highest density color suggests the (currently) most preferred plan as shown in figure 6(b);
- Plans with same high density color suggests that the user wavered in his/her preference as shown in figure 6(c);
- The distribution of the plans with high density color suggests the extent of the concern of the user.

EXPERIMENTS

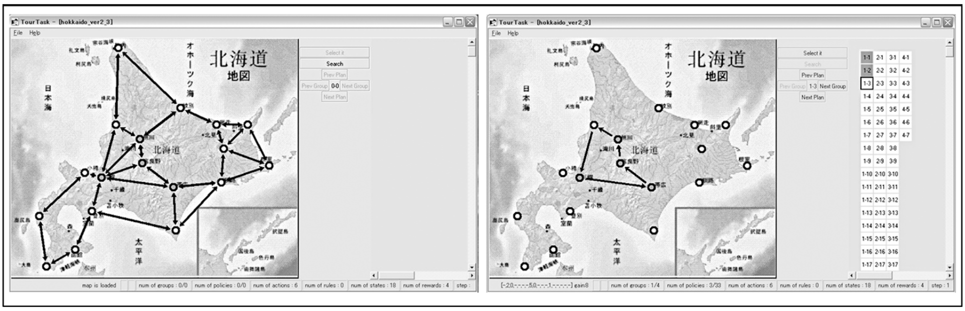
This section first describes the experimental setup. To examine the effects our interactive recommendation system, we perform the comparative experiments. There are two objectives. First one is "dose the preference of a user change during recommendation process?" Second question is "what kind of effect occurs by our awareness based recommendation?"

A total of sixteen subjects are divided into two groups for comparative experiment. Figure 7 shows the experimental condition whether recommendation space is visualized or not during the experiment. Both no visualizing group and visualizing group are eight subjects each. During the interactive recommendation as described above, a subject can select either coarse recommendation step or fine recommendation step until the subject decide the most preferred plan.

Solutions and Recommendations: Experimental Results

1. Does the preference of a user change during recommendation process?

Figure 7. The experimental condition on the recommendation space



(a) No visualizing condition

(b) Visualizing condition

Table 2 shows the number of subjects whether the preference has been changed or not. As the result, about 70% of subjects (11 subjects among 16 subjects) have changed their preferences. There is no significant difference whether visualizing or not. This result suggests that our interactive recommendation method brings out the user's preference change.

To evaluate the awareness support effect, we analyze the preference changes of each user in Table 2. Figure 8 shows an illustrated example of the distribution of the preference changes. In the graph, there are three kinds of round-trip plans. In each plan, four black circles are the goal cities that the user wants to visit, a white circle in a round-trip plan is the additional goal city that the user wants to become to visit during the plan recommendation. In the graph, horizontal axis is the number of preference extension which is the number of additional goal cities of the round-trip plan, and vertical axis is the number of preference shift which is the number of reduced goal cities from the round-trip plan. In Figure 8, (a) shows an example of the optimal plan which includes all four goal cities with the minimum plan length, (b) shows an example of the preference extended plan which includes all four goal cities with nine additional goal cities, and (c) shows an example of the preference shifted plan which includes only one goal city with one additional city.

Figure 9 shows the distribution of the preference changes of all subjects. Figure 9(a) shows the distribution of the preference changes of no visualizing condition group, and figure 9(b) shows the distribution of the preference changes of visualizing condition group. In each graph, horizontal axis is the number of preference extension, vertical axis is the number of preference shift, and the dotted circle covers the distribution of subjects who change the preference. Compared with the degree of preference changes between two conditions, visualizing condition group in figure 9(b) has larger preference changes than no visualizing condition group in figure 9(a) in both the preference shift axis and the preference extension axis. This suggests that our awareness support by visualizing condition works more effectively for assist preference change.

Table 2. The number of subjects with preference change

Condition	Result	
	Preference Change	No Preference Change
No visualizing	6	2
Visualizing	5	3
Total	11	5

Figure 8. Illustrated example of preference changes

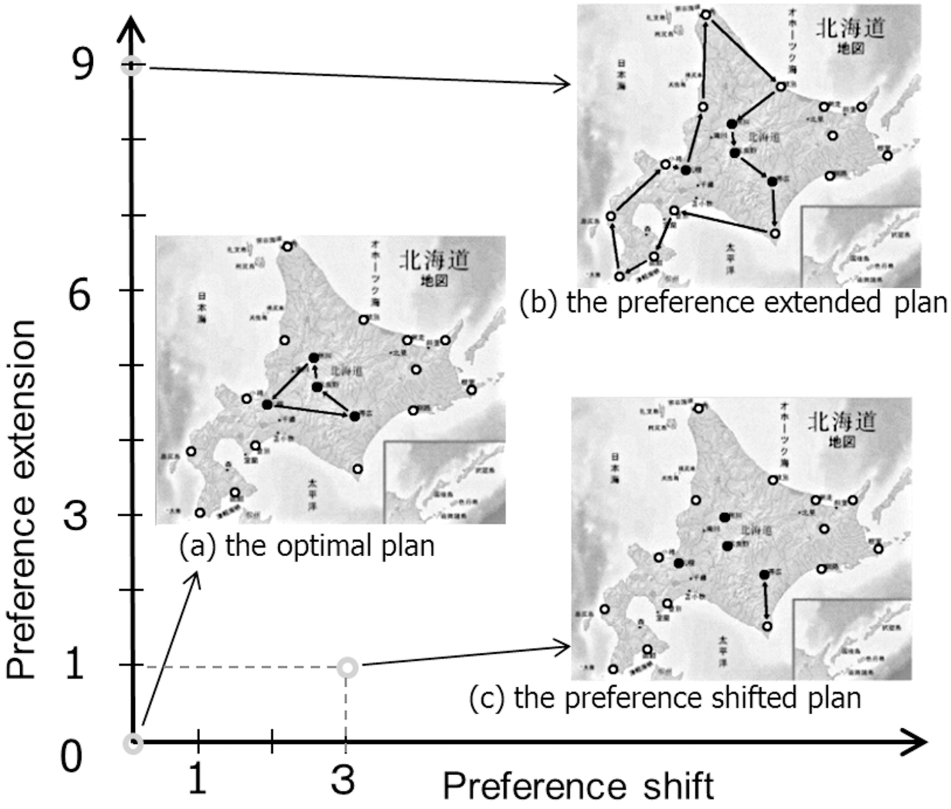
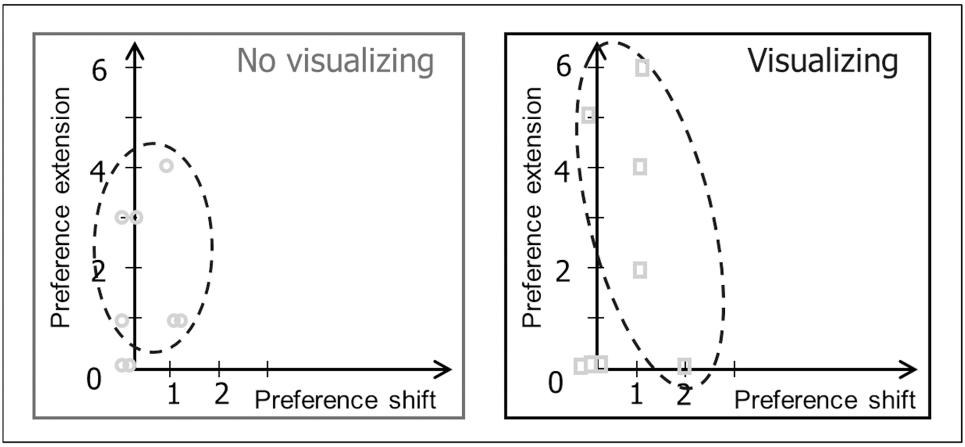


Figure 9. The distribution of the preference changes of all subjects.



(a) No visualizing condition *(b) Visualizing condition*

2. What kind of effect occurs by our awareness based recommendation?

To look into the effects of our recommendation, we analyze the subjects' behavior. Figure 10 shows the rate of referred plans during the interactive recommendation. Horizontal axis is ID of sixteen subjects. What's interesting is that 16 subjects are divided into two groups whether almost all plans have been referred or not. One group of 8 subjects is called *heavy users* who decide the most preferred plan after watching almost all plans. Another group of 8 subjects is called *light users* who do not watch all plans since they stop watching when a preferred plan is found.

To conduct the comparative analysis between heavy users and light users, we focus on the reference behaviors between them. Figure 11 shows the correlation between the average reference time per referred plan and the average number of references per referred plan. Vertical axis is the average number of references per referred plan, which is the total number of watched plans divided by the different kinds of watched plans. Horizontal axis is the average reference time per referred plan, which is the total reference time divided by the total number of watched plans. Among heavy users, there is a strong positive correlation (0.72) between the average reference time per referred plan and the average number of references per referred plan. Heavy users who watches almost all recommended plans tend to watch each plan quickly. Contrast to it, among light users, there is a negative correlation (-0.44) between the average reference time per referred plan and the average number of

Figure 10. The rate of referred plans during the interactive recommendation

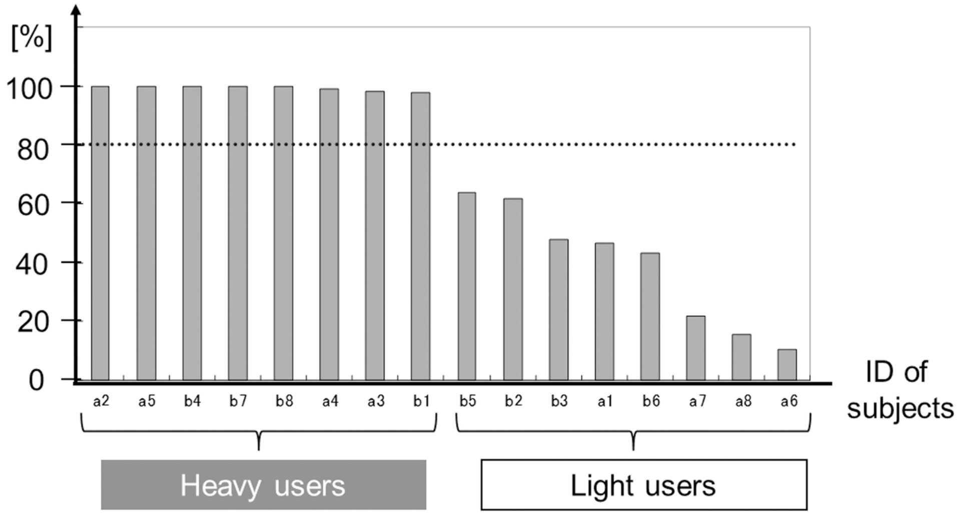
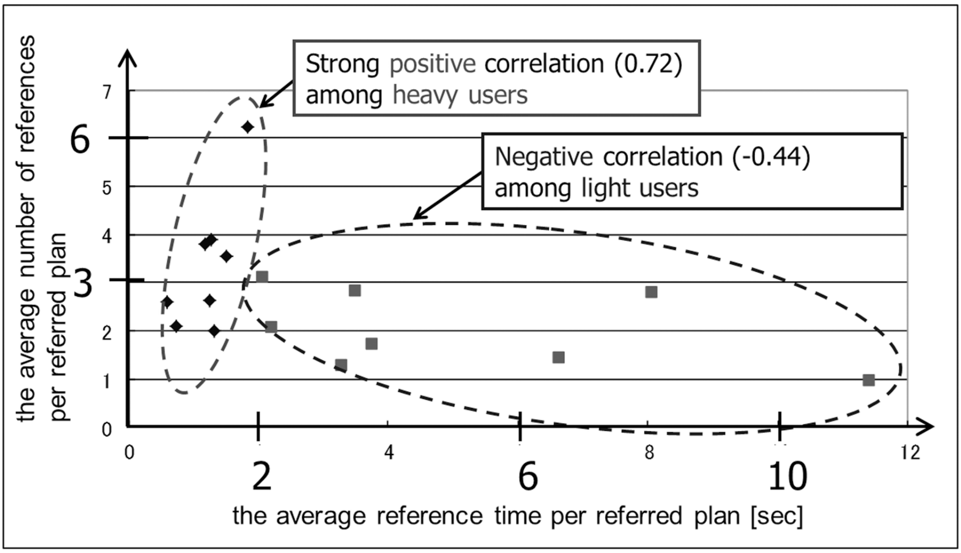


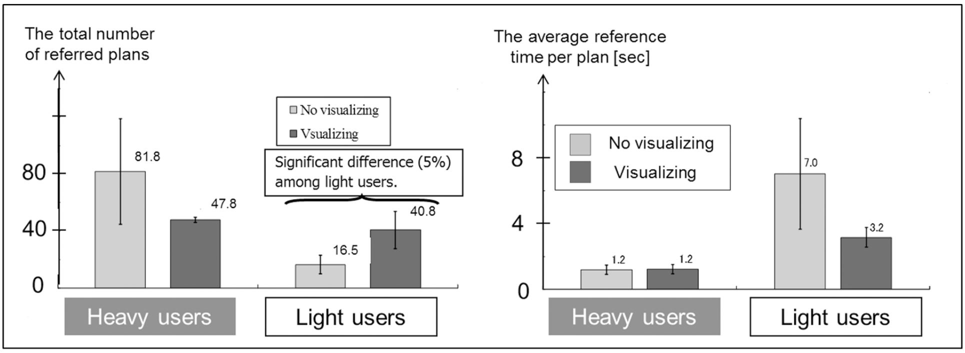
Figure 11. The correlation between the average reference time per plan and the average number of reference per referenced plan



references per referred plan. Light users who watches a part of recommended plans tend to watch his/her preferred plan for a long time. The reason is discussed later.

After this, figure 12 shows the comparative analysis in each heavy users and light users. Figure 12(a) shows the total number of referred plans between heavy users and light users. Vertical axis is the total number of references that is the number of watched plans. Horizontal line is classified result (heavy users or light users) and experimental conditions whether visualizing or not in each group. What's interesting

Figure 12. The comparative analysis in each heavy users and light users



(a) The total of referred plans *(b) The average reference time*

is that by visualizing, heavy users reduced the number of reference. It suggests that heavy users improve the efficiency by visualizing support. Besides, the number of reference between heavy users and light users is homogenized by visualizing condition since the light users increase the number of reference. There is significant difference (5%) among light users.

Figure 12(b) shows the average reference time per plan between heavy users and light users. Horizontal axis is same as figure 12(a). Vertical axis is the average reference time per plan. What's interesting is that by visualizing, light users reduced the average reference time. It suggests that light users improve the efficiency by visualizing support. On the other hand, there is no difference in heavy users in this metric. Besides, the average reference time between heavy users and light users is slightly homogenized by visualizing condition.

3. Summary of visualizing effects.

Now we summarize the visualizing effects on our experiments. First, subjects are classified into two groups, heavy users and light users by their behaviors. Second, different effects occurred by visualizing condition. Heavy users reduced the number of reference and the total reference time. On the other hand, light users reduced the average reference time, and increased the number of reference. Third, both group improved the efficiency for deciding the most preferred plan.

DISCUSSIONS

1. What is the major difference between heavy users and light users? First we discuss the major difference between heavy users and light users. Compare the average reference time per referred plan between them as shown in Figure 11, the heavy users tend to watch each plan more quickly than the light users. The major reason is that a user who has the standard of his/her preference to select the round-trip plan can watch each plan quickly. Contrast to it, a user who does not have the standard of his/her preference takes a long time to watch the round-trip plan since his/her is looking for the preferred way to select the plans during watching them. The research on from Novice to Experts, we show findings of the major difference between them. Bransford et al. (2000) describe several key principles of experts' knowledge in chapter 2 "How Experts Differ from Novices" as follows:

   a. Experts notice features and meaningful patterns of information that are not noticed by novices.
   b. Experts are able to flexibly retrieve important aspects of their knowledge with little attentional effort. These findings suggest that the major difference between heavy users and light users is same as the major difference between experts and novices.
2. Significance of the user-centered recommendation. Next we discuss the major significance of the user-centered recommendation. A main point is the way to bring out human awareness. First issue is "why an intelligent agent seems to be officious?" An agent is active and less intelligent than a human. The aspect of less intelligence can cause trouble to catch the human needs. Then the human might feel officious since the agent is active but needless. Second issue is "why passive recommendation is better?" The aspect of passive interaction results in ability for reflecting human intelligence. It covers less intelligence of the agent. Besides, it also brings out human awareness. An agent is active and less intelligent than a human. The aspect of less intelligence can cause trouble to catch the human needs. Then the human might feel officious since the agent is active but needless. Second issue is "why passive recommendation is better?" The aspect of passive interaction results in ability for reflecting human intelligence. It covers less intelligence of the agent. Besides, it also brings out human awareness.
3. Awareness for the preference distribution. Now we discuss the effect of awareness by visualizing the distribution of the user's preference trace in the recommendation space. We present two kinds of awareness by coarse to fine recommendation as follows:
   a. **Aware of the Preference Reduction by Coarse Axis:** The distribution toward coarse axis of the user's preference trace suggests the shift of the preference. Since the leftmost Group1 is every-visit plans including all original goals, when the most preferred plan is in the Group1, the preference of the user dose not shifted by the recommendation. On the other hand, if the most preferred plan is far from the Group1 as the right side to coarse axis, the preference of user has shifted during the recommendation process;
   b. **Aware of the Preference Extension by Fine Axis:** The distribution toward fine axis of the user's preference trace suggests the extension of the preference. Since fine axis vertically means the degree of plan length, the upper side plans are short, in other words, the number of cities without goals is small in each group. Contrast to it, the lower side of plans is longer with many cities to be visit. When the most preferred plan is in the upper side, the preference of the user dose not been extended by the

recommendation. On the other hand, if the most preferred plan is lower side toward fine coarse axis, the preference of user has been extended during the recommendation process since many alternative cities except goal cities are included.

4. Awareness and Reflection in human learning processes.

This section briefly looks at awareness process. First we summarize the recent research on designing awareness process as pre-learning process for reinforcement learning proposed by Yamaguchi et al. (2015), then survey researches on awareness and reflection in human learning processes. To formalize continuous leaning processes proposed by Buckler (1996) for both human and robot learning, Yamaguchi et al. (2013) focus on both the reinforcement learning framework for the learning agent and the continuous learning model of a human. To fill in the missing piece of reinforcement learning whose learning process is mainly behavior change, we add two mental learning processes, awareness as pre-learning process and reflection as post-learning process. Next, we survey researches on awareness and reflection to focus on the meaning of them by Yamaguchi et al. (2015).

In the research field of Technology-Enhanced Learning (TEL) for human learners, *Awareness* and *Reflection* (AR) is one of the active research issues on AR-TEL 2012 by Moore et al. (2012). 2013 year's theme for Workshop on this field, AR-TEL 2013 by Kravcik et al. (2013) is: "How can awareness and reflection support learning in different settings (work, education, continuing professional development, lifelong learning, etc.). What are the roles that technology can play to support awareness and reflection in these contexts?" We refer our meanings of *Awareness* and *Reflection* from these workshops papers.

The common feature of *Awareness* and *Reflection* is focusing experience of a learner on some information for future improvement. The main differences between them are that *Awareness* relates to the perception, Reflection mainly relates to the action or the behavior consisting of perception and action.

i. Awareness as pre-learning process.

There are several meanings of awareness on learning. Closely related meanings to our research are follows:

- Kurapati et al. (2012) discuss that individual situational awareness is defined as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future";

- Reinhardt & Christian (2011) discuss that awareness is increasingly related to finding appropriate learning objects, peers and experts or the 'right' learning path.

As pre-learning process, awareness plays an important part to trigger behavior change. Then we focus on the meaning of *awareness* as the need for distinction of indistinguishable perceptions or experience between future success and future failure. We assume that these indistinguishable perceptions occur by partially observable states discussed by Kaelbling et al. (1998).

ii. Reflection as post-learning process.

There are several meanings of reflection on learning. One of them is as follows; Learning through reflection is defined by Boud et al. (1985) as "those intellectual and affective activities in which individuals engage to explore their experiences in order to lead to new understandings and appreciations". We take to the concept of *Reflective learning* by Krogstie & Prilla (2012) and Krogstie et al. (2013) which is "the conscious re-evaluation of experience for the purpose of guiding future behavior". As post-learning process, reflection plays a part to create some meaning out of behavior change or learning result. Then we focus on the meaning of reflection as creating the becoming explanation or interpretation toward the rule which can lead the learner not to future failure but to future success.

FUTURE RESEARCH DIRECTIONS

This section describes introducing probabilistic events in the trip plan recommendation for future research directions. *A probabilistic event* is an event that occurs probabilistically, such as aurora-watching or cherry-blossom viewing (Hanami in Japan). To introduce a probabilistic event and its information into the plan recommendation, there are several issues. First issue is a kind of probabilistic events, we define two kinds of probabilistic events. Table 3 shows classifying the probabilistic events and their examples. *A probabilistic sightseeing event* is that the event to be watched which occurs probabilistically, such as aurora-watching or cherry-blossom viewing. *An additional probabilistic event* is the probabilistic event that affects other event such as the weather affects an outdoor event.

Second issue is how to estimate the probability of such event. A typical probabilistic sightseeing event has a season. There are on-season and off-season. So this paper estimates the occurrence probability of an event *P* as follows;

Table 3. Classifying the probabilistic events and their examples

Class	Example
Probabilistic sightseeing event	● Natural phenomenon: Aurora-watching ● Animals and plants: Cherry-blossom viewing
Additional probabilistic event	● Weather: Non-rainfall day

$$P\left(\text{event}\right) = \frac{\text{the number of occurrence days of the event}}{\text{the number of on season days of the event}}$$

1. The probabilistic sightseeing events. Then, we describe two examples of *the probabilistic sightseeing events*.
    a. Aurora-watching in summer at Yellowknife. Although, generally the main season of aurora-watching is winter, aurora-watching in summer is available at Yellowknife in Canada. Franklin House Marketing (2017) gives out information about the record of aurora-watching date. Referring to this record, aurora-watching is available for 44 days out of 61 days in September and October in 2015. Therefore, its occurrence probability is about 44/61 $\approx$ 0.72.
    b. Cherry-blossom viewing in Hokkaido, Japan. We discuss cherry-blossom viewing in Hokkaido cities as the probabilistic event. Table 4 shows the cherry flowering date from 2011 to 2017 years by Japan Meteorological Agency (2017a) at major four cities in Hokkaido. In Table 4, the cherry flowering date goes north from Hakodate (south end in Hokkaido) to Wakkanai (north end). So the possible season of cherry-blossom viewing in Hokkaido is from April 24 (at Hakodate) to June 10 (at Wakkanai, 14 days after May 28), for 48 days. Note that the number of cherry flowering date (from flowering to falling cherry blossoms) is about 14 days (two

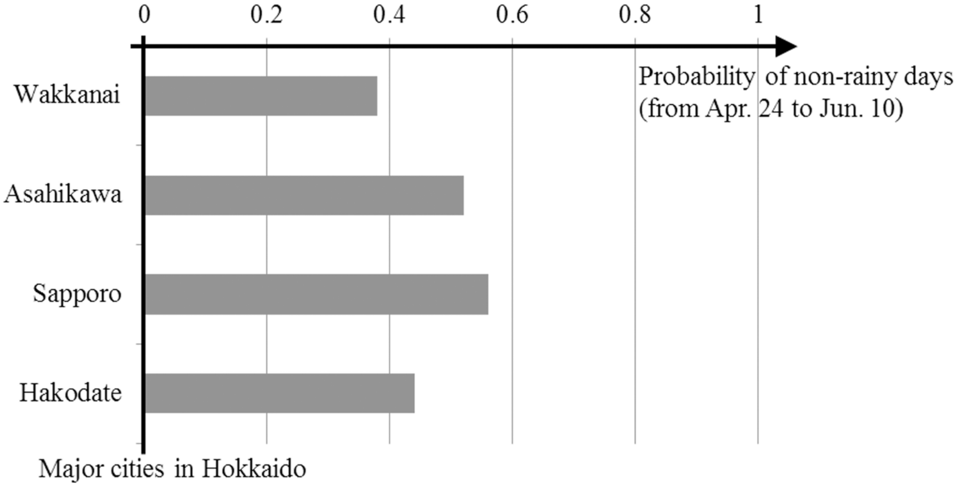
Table 4. Cherry flowering date (from 2011 to 2017 years) at major four cities in Hokkaido

Area	City	Cherry Flowering Date			Approximate Number of Flowering Days
		Earliest	Latest	Normal	
North end	Wakkanai	May 6	May 28	May 17	14 days
Middle area	Asahikawa	April 28	May 20	May 7	
	Sapporo	April 26	May 17	May 7	
South end	Hakodate	April 24	May 12	May 4	

weeks) in Japan. Then we estimate the probability of the cherry-blossom viewing event in Hokkaido for 48 days (from April 24 to June 10) is 14/48 $\approx$ 0.29.

2. The additional probabilistic event. Next we introduce the additional probabilistic event to affect other event. The role of the additional probabilistic event is to modify the probability of other event. Typical additional probabilistic event is weather such as whether a rainy day or not. In outdoor event, weather is very important to enjoy it. In cherry-blossom viewing event, weather condition is very important for travellers since it needs the provision of cold and rain falls. Then we describe the way to estimate the probability of weather event. Japan Meteorological Agency (2017b) gives out the weather data of major cities in Japan. Figure 13 shows the estimated probability of non-rainy days during cherry-blossom flowering days for major four cities in Hokkaido. As shown in figure 13, Sapporo is the best city to visit for cherry-blossom viewing since the probability of non-rainy is biggest (about 0.56). Note that there are many rainy days in Japan in cherry-blossom viewing season.
3. The plan recommendation for probabilistic event. Finally, we discuss the basic plan recommendation with probabilistic events. For a light user, a reliable plan with high occurrence probability event is recommended. Contrast to it, a plan with rare events is for a heavy user. To cover its rarity, the system also presents the recommended length of stay to meet the rare event certainly. For example, the length of stay for the existing aurora-watching tour is three nights. The

Figure 13. The probability of non-rainy days during cherry-blossom flowering days



reason is that the probability to occur the aurora-watching event for one night or more is $1-(1-0.72)^3 \approx 0.98$. This recommendation with occurrence probability of the event supports a user to decide the length to stay according to the degree of his/her interestingness.

CONCLUSION

We described awareness based recommendation by passively interactive learning system. For considering preference shift of a user, we proposed user-centered interactive recommendation by visualizing both the recommendation space with prepared recommendation plans and the user's preference trace as the history of the recommendation in it. Experimental results showed that subjects are divided into two groups, heavy users and light users, and that there are different effects between heavy users and light users under the same visualizing condition. Heavy users reduced the number of reference and the total reference time. On the other hand, light users reduced the average reference time, and increased the number of reference. They also showed that our system improves the efficiency for deciding the most preferred plan for both heavy users and light users. As future research directions, a probabilistic event and its basic recommendation way are discussed. Future work is ways to help to widen light users' perspectives through being exposed to different values.

ACKNOWLEDGMENT

The authors would like to thank Prof. Shimohara and Prof. Habib for offering a good opportunity to present this research. The authors would also like to thank Maiko Adachi, Kouki Takemori, Tatsuya Muraoka and Yuki Tamai for helping prepare the paper. This work was supported by JSPS KAKENHI (Grant-in-Aid for Scientific Research (C)) Grant Number 23500197.

REFERENCES

Anderson, N. H., & Butzin, C. A. (1974). Performance = Motivation X Ability: An Integration-Theoretical Analysis. *Journal of Personality and Social Psychology*, *30*(5), 598–604. doi:10.1037/h0037447

Balabanovic, M., & Shoham, Y. (1997). Fab: Content-Based, Collaborative Recommendation. *Communications of the ACM*, *40*(3), 66–72. doi:10.1145/245108.245124

Boud, D., Keogh, R., & Walker, D. (1985). *Reflection: Turning experience into learning*. Routledge.

Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.). (2000). *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*. Washington, DC: National Academy Press.

Buckler, B. (1996). A learning process model to achieve continuous improvement. *The Learning Organization*, *3*(3), 31–39. doi:10.1108/09696479610119660

Franklin House Marketing. (2017). *Canada Aurora Visitor Information.* Retrieved July 24, 2017, from http://www.canadaauroranetwork.com/

Griffith, S., Subramanian, K., Scholz, J., Isbell, C. L., & Thomaz, A. (2013). Policy Shaping: Integrating Human Feedback with Reinforcement Learning. In *Proceedings of 27th Annual Conference on NIPS 2013* (pp. 2625-2633). La Jolla, CA: Neural Information Processing Systems Foundation, Inc.

Habib, M. K. (2006). Mechatronics Engineering: The Evolution, the Needs and the Challenges. In *Proceedings of the 32nd Annual Conference of the IEEE Industrial Electronics Society, IECON 2006* (pp. 4510-4515). New York, NY: Institute of Electrical and Electronics Engineers (IEEE). 10.1109/IECON.2006.347925

Habib, M. K. (2007). Mechatronics: A Unifying Interdisciplinary and Intelligent Engineering Paradigm. *IEEE Industrial Electronics Magazine*, *1*(2), 12–24. doi:10.1109/MIE.2007.901480

Japan Meteorological Agency. (2017a). *Cherry flowering date (from 2011 to 2017 years).* Retrieved July 21, 2017, from http://www.data.jma.go.jp/sakura/data/sakura003_06.html

Japan Meteorological Agency. (2017b). *Search past weather data.* Retrieved July 25, 2017, from http://www.data.jma.go.jp/obd/stats/etrn/index.php

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*(1-2), 99–134. doi:10.1016/S0004-3702(98)00023-X

Kaplan, F., Oudeyer, P.-Y., Kubinyi, E., & Miklosi, A. (2002). Robotic clicker training. *Robotics and Autonomous Systems*, *38*(3-4), 197–206. doi:10.1016/S0921-8890(02)00168-9

Konda, T., Tensyo, S., & Yamaguchi, T. (2002a). LC-Learning: Phased Method for Average Reward Reinforcement Learning: Analysis of Optimal Criteria. In M. Ishizuka & A. Sattar (Eds.), *PRICAI2002: Trends in Artificial Intelligence, Lecture notes in Artificial Intelligence 2417* (pp. 198–207). Berlin, Germany: Springer. doi:10.1007/3-540-45683-X_23

Konda, T., Tensyo, S., & Yamaguchi, T. (2002b). Learning: Phased Method for Average Reward Reinforcement Learning (Preliminary Results). In M. Ishizuka & A. Sattar (Eds.), *PRICAI2002: Trends in Artificial Intelligence, Lecture notes in Artificial Intelligence 2417* (pp. 208–217). Berlin, Germany: Springer. doi:10.1007/3-540-45683-X_24

Konidaris, G., & Barto, A. (2006). Automonous Shaping: Knowledge Transfer in Reinforcement Learning. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 489-496). New York, NY: ACM

Kravcik, M., Krogstie, B., Moore, A., Pammer, V., Pannese, L., Prilla, M., . . . Ullmann, T. (Eds.). (2013). In *Proceedings of the 3rd Workshop on Awareness and Reflection in Technology Enhanced Learning (ARTEL2013), in conjunction with the 8th European Conference on Technology Enhanced Learning(EC-TEL2013).* Retrieved from http://ceur-ws.org/Vol-1103/

Krogstie, B., & Prilla, M. (2012). Tool support for reflection in the workplace in the context of reflective learning cycles. In *Proceedings of the 2nd Workshop on Awareness and Reflection in Technology-Enhanced Learning (ARTEL 2012)* (pp. 57-72). Academic Press.

Krogstie, B., Prilla, M., & Pammer, V. (2013). Understanding and Supporting Reflective Learning Processes in the Workplace. In *Proceedings of the Eighth European Conference on Technology Enhanced Learning (EC-TEL 2013), LNCS* (vol. 8095, pp. 151-164). Springer.

Kurapati, S., Kolfschoten, G. L., Verbraeck, A., Drachsler, H., Specht, M., & Brazier, F. M. T. (2012). A Theoretical Framework for Shared Situational Awareness in Sociotechnical Systems. In *Proceedings of the 2nd Workshop on Awareness and Reflection in Technology-Enhanced Learning (ARTEL 2012)* (pp. 47-53). Academic Press.

Mahadevan, S. (1996). Average Reward Reinforcement Learning: Foundations, Algorithms, and Empirical Results. *Machine Learning*, *22*(1-3), 159–195. doi:10.1007/BF00114727

Marthi, B. (2007). Automatic shaping and decomposition of reward functions. In *Proceedings of the 24th international conference on Machine learning* (pp. 601-608). New York, NY: ACM. 10.1145/1273496.1273572

Moore, A., Pammer, V., Pannese, L., Prilla, M., Rajagopal, K., Reinhardt, W., . . . Voigt, C. (Eds.). (2012). *Proceedings of the 2nd Workshop on Awareness and Reflection in Technology Enhanced Learning (ARTEL2012), in conjunction with the 8th European Conference on Technology Enhanced Learning(EC-TEL2012).* Retrieved from http://ceur-ws.org/Vol-931/

Ng, A. Y., Harada, D., & Russell, S. J. (1999). Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *Proceedings of the 16th International Conference on Machine Learning* (pp. 278-287). New York, NY: ACM

Ng, A. Y., & Russell, S. (2000). Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML2000)* (pp. 663-670). San Francisco, CA: Morgan Kaufmann Publishers Inc.

Peterson, G. B. (2000). The Discovery of Shaping or B.F. Skinner's Big Surprise. *The Clicker Journal: The Magazine for Animal Trainers*, *43*, 6–13.

Peterson, G. B. (2001). The world's first look at shaping: B. F. Skinner's gutsy gamble. *The Clicker Journal: The Magazine for Animal Trainers*, *2001*, 14–21.

Pryor, K. (2006). *Don't Shoot the Dog! The New Art of Teaching and Training*. Lydney, UK: Ringpress Books.

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY: John Wiley & Sons, Inc. doi:10.1002/9780470316887

Reinhardt, W., & Christian, M. (2011). Awareness in Learning Networks. In *Proceedings of the PLE Conference 2011 (ARTEL2011)* (pp. 12-20). Academic Press.

Riecken, D. (2000). Introduction: Personalized views of personalization. *Communications of the ACM*, *43*(8), 26–28. doi:10.1145/345124.345133

Satoh, K., & Yamaguchi, T. (2006). Preparing various policies for interactive reinforcement learning. In *Proceedings of the SICE-ICASE International Joint Conference 2006 (SICE-ICASE 2006)* (pp. 2440-2444). New York, NY: Institute of Electrical and Electronics Engineers (IEEE). 10.1109/SICE.2006.315139

Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-Commerce Recommendation Applications. *Journal of Data Mining and Knowledge Discovery*, *5*(1/2), 115–153. doi:10.1023/A:1009804230409

Takadama, K., Sato, F., Otani, M., Hattori, K., Sato, H., & Yamaguchi, T. (2012). Preference clarification recommender system by searching items beyond category. In *Proceedings of IADIS (International Association for Development of the Information Society) International Conference Interfaces and Human Computer Interaction 2012* (pp. 3-10). Lisbon, Portugal: IADIS Press.

Yamaguchi, T., Nishimura, T., & Sato, K. (2011). How to recommend preferable solutions of a user in interactive reinforcement learning? In A. Mellouk (Ed.), *Advances in Reinforcement Learning* (pp. 137–156). Rijeka, Croatia: InTech Open Access Publisher. doi:10.5772/13757

Yamaguchi, T., Nishimura, T., & Takadama, K. (2009). Awareness based filtering - Toward the Cooperative Learning in Human Agent Interaction. In *Proceedings of the ICROS-SICE International Joint Conference (ICCAS-SICE 2009)* (pp. 1164-1167). Tokyo, Japan: The Society of Instrument and Control Engineers.

Yamaguchi, T., Nishimura, T., & Takadama, K. (2012). *Awareness based recommendation - Toward the Cooperative Learning in Human Agent Interaction*. Paper presented at The International Conference on Humanized Systems 2012 (OS02_1003). Daejeon, South Korea.

Yamaguchi, T., Takemori, K., & Takadama, K. (2013). Modeling a human's learning processes toward continuous learning support system. In M. K. Habib & J. P. Davim (Eds.), *Mechatronics Engineering* (pp. 69–94). Hoboken, NJ: Wiley-ISTE.

Yamaguchi, T., Takemori, K., Tamai, Y., & Takadama, K. (2015). Analyzing human's continuous learning processes with the reflection sub task. *Journal of Communication and Computer*, *12*(1), 20–27.

KEY TERMS AND DEFINITIONS

Additional Probabilistic Event: It is the probabilistic event that affects other events.

Awareness-Based Recommendation: It is the user-centered recommendation by visualizing both the recommendation space with prepared recommendation plans and the user's preference trace as the history of the recommendation in it. The recommendation space visualizes the possible preference shift of the user.

Heavy Users: Users of the interactive recommendation system who decide the most preferred plan after watching almost all plans.

Human Adaptive and Friendly: Less active but more intelligent agent is desirable since it does not seem to be officious for the human.

Interactive Recommendation Space: In the recommendation space, the user can view and select various plans actively. The recommendation space consists of two dimensions; the preference reduction axis and the preference extension axis, in that, various plans are arranged in a plane.

Interactive Reinforcement Learning With Human: Reinforcement learning method in that reward function denoting goal is given interactively by a human. It is not easy to keep reward function being fixed while the human gives rewards for the reinforcement learning agent. It is that the reward function may not be fixed for the learning algorithm if an end-user changes his/her mind or his/her preference.

Light Users: Users of the interactive recommendation system who do not watch all plans since they stop watching when a preferred plan is found.

Model of a User's Preference Shift: It is defined by two axes, preference reduction and preference extension. Comparing the previous preference set and the current preference set, the common set is the invariant preference, the reduction set is called preference reduction, and the addition set is called preference extension.

Preference Change Problem: It is a problem that the collected user's profile is not same as the user's current preference.

Probabilistic Event: It is an event that occurs probabilistically, such as aurora-watching or cherry-blossom viewing.

Visualizing the Recommendation Space: The objective of this visualization is to inform a user of two kinds of information. First is that the recommendation space consists of two-axes. Second is that in each axis, groups or plans are ordered according to the recommendation order.

Visualizing the User's Preference Trace: The objective of this visualization is to show the distribution and the degree of the user's preference to him/herself.

Chapter 10
Fusion of Gravitational Search Algorithm, Particle Swarm Optimization, and Grey Wolf Optimizer for Odor Source Localization

Upma Jain
Atal Bihari Vajpayee Indian Institute of Information Technology and Management, India

Ritu Tiwari
Atal Bihari Vajpayee Indian Institute of Information Technology and Management, India

W. Wilfred Godfrey
Atal Bihari Vajpayee Indian Institute of Information Technology and Management, India

ABSTRACT

This chapter concerns the problem of odor source localization by a team of mobile robots. A brief overview of odor source localization is given which is followed by related work. Three methods are proposed for odor source localization. These methods are largely inspired by gravitational search algorithm, grey wolf optimizer, and particle swarm optimization. Objective of the proposed approaches is to reduce the time required to localize the odor source by a team of mobile robots. The intensity of odor across the plume area is assumed to follow the Gaussian distribution. Robots start search from the corner of the workspace. As robots enter in the vicinity of plume area, they form groups using K-nearest neighbor algorithm. To avoid stagnation of the robots at local optima, search counter concept is used. Proposed approaches are tested and validated through simulation.

DOI: 10.4018/978-1-5225-5276-5.ch010

INTRODUCTION

Multi Robot System

Multi robot system can be defined as a group of homogeneous or heterogeneous robots working together to achieve the desired goal. The ability of a multi-robot system to detect, locate or follow a target either in known or unknown area, has various valuable applications in real life. Applications of target searching include search and rescue after disaster, toxic gas detection, narcotics or explosives localization and so on. For instance, the use of autonomous robots for search task in dangerous environment has been receiving consistent attention. Searching using a team of mobile robots, instead of a single robot has potential advantages (viz. fault tolerance, accuracy) and also has numerous challenges (viz.coordination, communication, collision avoidance.

Target Searching

The problem of target localization concerns with the navigation of one or more robot towards the target source. Target may be static or mobile, intensity of plume may be constant or plume may vary, number of target may be one or many. Depending on the target a broader spectrum of target locating problems have emerged ranging from odor source localization (Jatmiko et al. 2011), target localization and tracking (Ramya et al. 2012), search and rescue (Liu, Y., & Nejat, G. 2013) and so on.

Odor Source Localization

Localization of an odor source can be seen as a behavioral problem which differs from animal to animal. Some animal senses the fluid concentration (lobster), some makes use of residues on the surface, while others can trace the plume in air (moths) or can use combination of information (dogs). Information of odor is widely used by animals for information exchange, finding mates and searching for food in nature. Advancement in the area of robotics and sensor technology has motivated researchers to use robots for odor source localization in environment harmful for human beings.

The process of odor source localization can be divided into three phases: odor plume finding, plume following and odor source declaration. During plume finding, robots keep on moving in workspace to encounter the plume area. Once the plume is find robots enter in the second phase i.e. plume following. In this phase robots follow the plume to find the odor source. Odor source declaration is the process of declaring that source has been found.

In order to make robots work, set of rules need to be defined. Such rules can be formulated in the form of an algorithm. Many species in the environment follow the traces of odor to find food or locate mate. Such behavior of species to follow the plume has been modeled in mathematical form to guide the robot towards the odor source. Most of the bio inspired algorithms are based on the behavior of moths. There are three kinds of behavior observed in moths namely upwind search (moves upwind when exposed to pheromone), casting (after losing the plume moving side to side in the hope of reacquiring the plume), spiraling (if still it can't find plume, starts circular motion).

Nature Inspired Method

Nature inspired algorithms find inspiration from nature. These algorithms are inspired from bio system, swarm behavior, chemical or physical system. Complexity of real world problems is increasing on everyday basis; optimization techniques that can find an acceptable solution within certain amount of time are required. Nature inspired algorithms have become a popular topic due to their ability to find near optimal solutions in reasonable time. Nature inspired paradigms have been used consistently in robotics for control, decision making and learning. Simple control rules drawn by the study of social insects and animals have been found suitable for multi-robot systems.

Particle Swarm Optimization, PSO (Eberhart, R. C., & Kennedy, J. 1995) proposed by kennedy and Ebarhart in 1995 is based on the social behavior of birds or fish schooling. Collective behavior of birds is modeled in terms of rules to obtain an optimized result. It is a computational approach to optimize problems by iteratively improving the candidate solution. In PSO two factors, particle's own experience and swarm's best solution are used to guide the particles towards the candidate solution. However PSO is likely to get stuck in local optima.

Gravitational Search Algorithm, GSA (Esmao Rashedi et al., 2009) works on the principle of newton law of gravity, according to which each particle have some attraction force towards every other particle. This force is governed by the masses of particles and distance among the particles. Each particle or agent is specified by position, active gravitational mass, passive gravitational mass and inertia. Particle's position corresponds to the solution. Masses are calculated by using the fitness function. In GSA, masses follow the law of gravity and motion. According to the law of gravity, amount of force among the particles is directly proportional to their masses and inversely proportional to the square root of distance between them. Law of motion states that, particle's new velocity is affected by, the particle's previous velocity and variations in current velocity. Variations in velocity are equivalent to the force acting on mass divided by inertia. GSA can be considered as the system of masses obeying the law of gravity and motion.

Grey Wolf Optimizer (GWO) (Mirjalili, S. et al. 2014) given by Seyedali Mirjalili et al. in 2014, is inspired from the behavior of grey wolves (Canis lupus). It mimics the leadership and hunting behavior of wolves. Grey wolves usually live in pack of 5-12 wolves. There are basically four hierarchical leadership positions namely alpha, beta, delta and omega. First level position is alpha. Alpha takes decision about sleeping place, hunting, time to wake etc. Alpha is best in terms of decision making not in terms of strength. All other wolves follow the alpha. Second hierarchal position is beta, beta helps alpha in decision making and make other wolves to follow the alpha command. Beta also passes information from wolves to alpha. Beta works as advisor to alpha. Third position delta wolves are subordinator wolves; they submit to alpha and beta but can dominate omegas. Scouts, caretakers, sentinels hunters belongs to this category. Last level is omega they are dominated by all three alphas, beta and delta.

Other than social hierarchy another behavior is hunting. First they chase and reach the prey then they encircle the prey and attacks until it stops moving. This hunting behavior and social hierarchy is mathematically modeled to design GWO.

This chapter discusses an approach which minimizes the time required to locate the odor source, by utilizing the strengths of GSA, GWO and PSO. The proposed method is the fusion of above three. GWO has been used in concatenation with the hybrid of GSA and PSO. First candidate solutions are created by the hybrid of GSA and PSO, and then resultant solutions are fed into GWO as initial solutions to get the final solutions and vice-versa.

This chapter is organized as follows: Section Related work provides a brief review of target searching methods. Section Methodology discusses the proposed target localization approach. Simulation setup details and results are discussed in section Simulation setup and Simulation results. Section Conclusion concludes the chapter with a discussion on possible future work.

RELATED WORK

Research on locating the odor source by using robots had begun in 1990s (Sandini etal. 1993). Early work comprise of single robot to localize the odor source. With the advancement in technology multi robot system has come into the existence to localize odor source. Many factors like environment, sensor response time, odor source model affects the process of odor source localization.

Problem of odor source localization requires knowledge of both fields, robotics and physics. Process of odor distribution is highly affected by environment. Such as whether the environment is indoor or outdoor, contains obstacle or not. Wind also plays an important role while finding the odor source. If the wind speed is high plume is carried out by the wind and becomes turbulence dominated and if wind speed is not high plume is said to be diffusion dominated. Along with the wind speed, wind stability also hinders the distribution of molecules around the environment.

Problem of odor source localization has already been studied by many researchers (Russell et al.2003, Marques et al. 2006, Jatmiko et al. 2011, Marjovi et al. 2011) Several methods have been proposed namely: Hex-path algorithm (Russell, R. A. 2003, Lilienthal etal. 2003), gradient following (Li et al. 2007), the zigzag path search strategy (Holland, O., & Melhuish, C. 1996), rule based strategy (Zarzhitsky et al. 2004), fluxotaxis (Gong et al. 2011), odor tracking (Vergassola et al. 2007), infotaxis (Kuwana, Y., & Shimoyama, I. 1998), combination of chemotaxis and anemotaxis (Hayes et al. 2002), cooperation based on swarm intelligence (Hayes et al. 2003).

Li et al. (Li et al. 2007) proposed a gradient-based method to locate the odor source. In this approach, robots make the use of concentration gradient information from other robots and apply global stochastic strategy to decide the next move. But the cooperation mechanism employed among robots is not sufficient to utilize the full information from other robots. Marques et al. (Marques et al. 2006) proposed a new PSO inspired cooperative algorithm which is based on the exchange of information among neighboring agents for searching odor sources across large search spaces. For spreading the agents around the search space a global searching behavior is integrated by taking into account repulsive forces among the agents and crosswind biased motion. Marques et al. (Marques et al. 2002) proposed a method for odor source localization by multiple robots. This method is based on genetic algorithms. Fitness of each robot is the measure of odor concentration. The algorithm increases the exploration capacity by applying crossover, mutation and selection operation. But convergence is slow if the initialization of robots is not random in search space. Hayes et al. (Hayes et al. 2003), presented a collaborative spiral surge approach on the basis of swarm intelligence and a control algorithm to localize the odor source by the team of autonomous robots. Marjovi and Marques (Marjovi, A., & Marques, L. 2011) proposed a cooperative approach to locate odor source in unknown structured environments. A frontier based decentralized approach has been used by considering the concentration of odor source and flow of air at each frontier to decide the next move. Topological graph matching techniques have been used for map merging. Jatmiko et al. (Jatmiko et al. 2009) proposed a cooperative approach for odor source localization based on nichie PSO. Performance is improved by dynamically adjusting the size of nichie. Zhang et al. (Zhang et al. 2013) proposed a multi-robot cooperation strategy based on the virtual physics force which incorporates four different forces, structure formation force, repulsion force, goal force and rotary force. A new method using rotary force is introduced to avoid the tracing of single target by more than one group of robots. Meng et al. (Meng et al. 2012) presented a cooperative search approach to realize odor source localization. The method uses the upwind search and ant colony algorithm at the same time. Couceiro et al. (Couceiro et al. 2011) proposed two approaches namely RPSO (Robotic PSO)and RDPSO (Robotic DPSO), based on the behavior of PSO and Darwinian Particle Swarm Optimization for multi-robot system and Odor source localization by GSA and GPSO algorithms 3 considered obstacle avoidance. The concept of punish reward mechanism is used to avoid the local optima. RDPSO is represented by multiple swarms, but how the swarm formation takes place has not been discussed. Jim Pugh and Alcherio Martinoli (Pugh, J., & Martinoli, A. 2007) proposed a search algorithm for multi-robots based on the principles of PSO. Authors achieved close matching among PSO and multi-robot system and also studied the effect of changing the number of robots and communication range on the performance of algorithm. PSO is studied with and without Global positioning system. Kurt Derr and Milos Manic (Derr, K., & Manic, M. 2009) proposed a distributed PSO based approach for search in high risk environment. A novel RSS weighing factor is used to avoid overshooting of the target. Performance is analyzed for single and multiple target case. Result demonstrates the effect of signal strength on the search time. Sheetal Doctor et al. (Doctor, S. et al. 2004) proposed an enhanced version of PSO. Performance of the algorithm is governed by parameters. Two level PSO has been used, inner level PSO is responsible to get the solution of the problem and the outer level is responsible for parameter tuning. This method has been tested for both single and multi-target case. In case of multi target number of targets available is assumed to be known previously. Kazadi et al. (kazadi S. et al. 2000) have given a simple approach to follow the water vapor plume. A mobile robot is used for tracing the vapor plume. It relies on boundary and wind information. Approach finds its strength in terms of low computational overhead and power consumption. Plume tracing has been done with different initial locations of robot. Result shows that effects of robot's initial placement are negligible on success rate. Willis (Willis, M. A. 2002) has given an approach for odor source localization. Approach is based on moth's behavior. If robot do not find odor plume it takes right angle turns across the wind line, otherwise it moves in the upwind direction to locate the odor source. Genetic algorithm has been used for parameter setting. Simulation has done to validate the approach. Ishida et al. (Ishida, H. et al. 1996) have given a new exploratory approach for locating the odor source. Algorithm makes use of concentration gradient information and wind information. If robot is in the plume area it progresses by following the concentration gradient. If plume is lost by robot it tries to relocate the plume by making move across the wind line. Real experiment was performed in a room to locate ethanol. Result shows the efficiency by locating ethanol almost from anywhere in room. Marques and Almeida (Marques, L., & De Almeida, A. T. 2000) have given an approach to find a specific odor source in presence of two odor sources. To identify the odor source neural network based pattern recognition algorithm is used. Plume tracking is done by three different algorithms. Result shows that the convergence rate of bacteria algorithm is slower in comparison of gradient and moth based algorithms. Gabriele et al. (Gabriele Ferri et al. 2009) have proposed a spiral search approach for odor source localization by an autonomous robot. Experiment was done in a weak airflow indoor environment. Robot progresses by moving along the spiral path and collects gas concentration information. This information is used to generate proximity index i.e. used to estimate the closeness of odor source. If robot finds out more promising position then it starts a new spiral otherwise continue to move in the same spiral. Experiment was done with MOMO platform robot. Results show the effectiveness of approach in indoor environment. Yuhua and Dehan (Yuhua Zou and Dehan Luo 2008) have proposed an approach based on ant colony optimization to locate odor source. Searching was done in three phases; global traversal, local traversal and pheromone update. In addition a verification process is added to find out multiple odor sources. Simulation results validate the performance of approach. Yong et al. (Yong Zhang et al. 2015) have given a refined hybrid approach by combining PSO with Bacterial Foraging Optimization (BFO) for odor source localization by a team of mobile robots. Two behaviors namely elimination dispersal and chemotaxis are adopted from BFO and integrated in PSO. Chemotaxis is integrated to help robots in plume following. Elimination dispersal is added to handle the problem of local optima during plume following. Simulation has done to validate the approach. Mohamed Awadalla et al. (Mohamed Awadalla et al. 2013) have presented a 3D framework for odor source localization by combining matlab and CFD. The process of fluid flow and contaminant transport in a virtual environment is described using CFD. This is a five step procedure, first the geometry is created, and then mesh is generated. After that generated mesh is gone through preprocessing, then solver is used to create solutions. Lastly post processing of results is done. Results are imported in matlab to provide concentration and airflow information. A virtual robot is used to test the performance of 2D and 3D algorithms. Performance of 2D algorithms is evaluated by providing concentration information at four specific heights one at a time. To evaluate the performance of 3D algorithms four sensors are placed vertically at different heights to provide concentration information. H_2S is taken as an odor source. Results show that 3D algorithms are more effective as compared to 2D algorithms. Kowadlo and Russell (G. Kowadlo and R. A. Russell 2006) have proposed an approach based on naïve physics for odor source localization. To develop airflow model, naïve physics is used. Idea is to avoid the complex differential equations by using common sense and physics for airflow model. To guide the robot naïve airflow model is used in conjunction with the map of environment. Possible locations of odor source are predicted by finding the airflow pattern. This approach do not need robot to move all across the plume for localizing the odor source. Approach is validated in known indoor environment. Jatmiko et al. (Jatmiko, W. et al. 2009) have proposed a ranged subgroup approach to find multiple odor sources by a team of mobile robots. A coordinator agent is used to manage niche. This coordinator agent is named as main robot. It is responsible for information sharing and handling robot membership. Niche is formed by the robots comes inside the attract boundary of main robot. To avoid following of same odor source by more than one niche, rejection character is adopted by main robot. Simulation has been carried out to show the validity of approach. Gong li et al. (Ji-Gong li et al. 2015) have presented an approach for odor sources mapping in a time variant airflow environment by using a mobile robot. For modeling the odor source, advection diffusion model is adopted. A dempster-Shafter theory based method is proposed to do the prediction about possible locations of odor source. Robot performs exploration in outdoor environment and collects information of airflow direction and odor concentration. Then this information is used by DS inference system for predicting the possible location of sources. Simulation results show the effectiveness of approach in predicting the possible locations of odor sources.

Various methods for target searching are proposed by researchers. Most of these methods are based on nature inspired algorithms due to their suitability for multi robot system. Among various nature inspired algorithms PSO has been applied extensively due to its simplicity, ease of implementation and promising results. For that reasons PSO is being used in our method too. One of our methods is the hybridization of PSO and GSA and second is based on GSA and third is the fusion of GSA, PSO and GWO.

METHODOLOGY

An efficient search algorithm balances both exploration of new areas and exploitation of high intensity plume areas. The Odor intensity around the odor source is modeled by assuming the odor intensity to follow a Gaussian distribution. This odor intensity across the search area is represented in the form of an image which comprises details of the odor concentration. The resultant image pixels have values ranging from 0 – 255. 0 corresponds to the area that does not contain any odor (plume concentration is zero) and pixels having value greater than 0 (threshold (Th1) value) is considered as plume vicinity area or active area. The objective of the robot is to locate the odor source which is assumed to be maximum pixel intensity area (greater than a certain threshold (Th2) (250)) of plume. These thresholds are arbitrarily assigned by assuming that the robot sensors to detect odor are perfect and that they can sense the plume intensity effectively. The process of odor source localization can divided into three steps such as plume finding, plume following and odor source declaration by the robot that reached the odor source. Search strategies used in the proposed algorithms can be broadly classified as global search and local search. In global search, the whole area is explored by a robot to locate the odor source. Once a robot finds an active area it forms a group with other robots using K-nearest neighbour (Mucherino, A. et al. 2009) algorithm. Local search is concerned with this second phase of odor source localization called as plume following. Local search is performed using GSA, Hybrid (GPSO) and fusion (GPSO-GWO and GWO-GPSO) approaches and the results are compared with PSO based search. The proposed approaches are found to outperform the PSO based search.

The overall procedure of odor source localization is described in the Main algorithm (Algorithm 1). The area is assumed to be rectangular and the robots are initialized to begin their search from one of the corners C_i. Robots continue to move randomly in the search space until at least one of the robots finds the active plume area (having the odor concentration greater than the threshold value 0). Once a robot finds the plume area (threshold > 0) it forms the group using k-nearest neighbor algorithm. Then the local search algorithm is used within this group to direct the robots to find the maximum intensity (odor concentration) area. Other robots keep exploring the area for other active areas and may form more groups based on if they had reached any active areas. Local search is performed until anyone of the robot of any group finds the odor source.

Initialization of Robots

Most of the nature inspired algorithm and optimization algorithms consider the random initialization of population across the search space. Random initialization of robots is not always realistic in multi-robot system. In our simulation work we assume the initial placement of robots at the corner of the search space.

Global search is performed around the whole workspace until at least one of the robots find the active plume area (having the odor concentration greater than the threshold value 0). During global search robots move randomly around the workspace to find the active plume area.

The team of robots is divided into groups by using k-nearest neighbor algorithm. In order to perform grouping of robots a list is generated. The authors calculate the fitness of each robot and if the fitness of robot is more than the threshold value (robot is in the vicinity of plume) then robot is placed in the active agent list. And the robot having highest fitness in active agent list forms the first group with the nearest two robots. Rest of the robots forms another group.

Algorithm 1: Main algorithm

```
Robots $R_i = (R_1, R_2 \ ..., R_6)$ initialization
Robots start global search from one of the corners $C_i$ where $C = \{C_1, C_2, C_3, C_4\}$ of the workspace
while (group is not formed or odor source has not been find) do
    For each robot calculate fitness
    fitness = intensity (odor concentration at that point) measured at time t
    if fitness > threshold (robot is in plume area) then
        Put robot in active agent list;
    end if
    if active agent list is not empty then
        Sort the list according to the maximum fitness value.
        Take first robot as the main robot call K (2)-Nearest neighbor algorithm to form the group
    else
        Search randomly, assign robot's next move randomly.
    end if
end while
if (Groups have formed)
    Do for each group
        Call local search algorithm for group1 and group2.
        I1 = fitness of best performing robot in group1.
        I2 = fitness of best performing robot in group2.
    while (I1>Th2 or I2>Th2).
end Do while
```

Algorithm 2: KNN based group formation

```
if active agent list is not empty then
    Choose first robot as main robot
    Calculate distance of every other robot from main robot.
    Select first two nearest robots from main robot.
    Main and nearest two robot will form the group1.
    Rest 3 robots forms the another group.
end if
```

Local Search Methods

After the formation of groups, robots exchange the information and perform the local search within the group. Local search aids the robots in tracing the plume to find the maximum intensity area. Local search is performed PSO based approach, GSA based approach, GPSO based search, Concatenated GPSO-GWO and GWO-GPSO.

PSO Based Search

Particle swarm optimization is an evolutionary algorithm inspired from social behavior of animals (birds, fishes). Among various nature inspired algorithms, PSO is used widely in various fields of robotics (obstacle avoidance, path planning, exploration, target searching). It is a population based search algorithm. Each particle in PSO is considered as a robot in a multi-robot system. In PSO the next position of the particle is found out by computing the global best particle among the population and particle's previous best position. Velocity refers to the speed at which the particle is moving. Velocity and position of the robot is updated by the equations 1 and 2.

$$v_k(t+1) = \mathrm{w} * v_k(t) + \mathrm{c}_1 * \ rand_1 * \left(pbest_k - x_k(t)\right) + \mathrm{c}_2 * \ rand_2 * \left(gbest - x_k(t)\right) \tag{1}$$

$$x_k\ (t+1) = x_k(t)\ +\ v_k\ (t+1) \tag{2}$$

where $v_k(t)$, $v_k(t+1)$ are the velocities of robot *k* at time *t* and *t* +1, w is the inertia weight. c_1, c_2 are positive constants. rand_1, rand_2 are random numbers which vary from 0 to 1. $pbest_{k}$, is previous best position of robot *k*, *gbest* is the position of best performing robot, x_k *(t)* and x_k (*t* + 1) are the positions of robot *k* at time t and *t* + 1.

Algorithm 3: PSO based search

```
Do
Evaluate each robot's fitness
fitness = intensity (odor concentration at that point) measured
Compare each robot's fitness.
Find gbest and pbest
$gbest(t) = \max\left(fit_k(t)\right), \forall k = 1,\ 2, \ldots\ldots, \mathrm{N}$
```

```
$pbest(t) = \max\left(fit_j(t), fit_j(t-1)\right)$
$fit_k(t)$: fitness of the robot $k$ at iteration $t$
5: Update velocity and position by equations 1 and 2.
6: While (odor source is not found)
```

GSA Based Search

Gravitational Search algorithm, GSA is introduced in 2009, which is based on Newton's law of gravity and motion. After its introduction, GSA has been applied extensively in many areas and modified accordingly. GSA is a population based algorithm where interaction among agents is performed based on by the law of gravity. In this work, GSA based search algorithm has been applied for multi-robot odor source localization. In the multi- robot system for odor source localization, each robot's position is defined by:

$$R_i = (R_1^1,\ R_1^2 ... R_1^d, R_2^1,\ R_2^2 ... R_2^d, ..., R_i^1,\ R_i^2 ... R_i^d) \tag{3}$$

$$R_i = (R_1^1,\ R_{2,}^d ... R_i^n) \tag{3}$$

where $d = 2$ and $i = 1$ to 6

R_i^d = position of robot i in dimension d

In GSA based odor source localization each robot is considered as a mass. Every robot is governed by its position, active and passive gravitational masses and inertial mass. The Mass of the robot depends on the fitness of robot and is calculated by the following equations

$$M_{as} = M_{ps} = M_{is} = M_{s,}\ s = 1, 2, ..., 6 \tag{4}$$

$$M_s(t) = \left(fit_s(t) - worst(t)\right) / \left(best(t) - worst(t)\right) \tag{5}$$

$$M_s(t) \ = m_s(t) \,/\, \Sigma m_t(t), \quad \forall\, t = \ 1,\ 2,\ldots,n \tag{6}$$

$fit_s(t):$ fitness value of robot *s* at iteration *t*.
$worst\,(t):$ fitness value of worst performing robot.
$best\,(t):$ fitness value of best performing robot at iteration *t*.
$M_{is}\ :$ inertia mass of the agent *s*.

At any particular time *t*, the force acting on a robot (agent) *s* from *t* is given by the equation

$$F_{st}{}^{d}(t) = \mathrm{G}\ (t) \times\ \Big(\ \big(M_{ps}(t)\ \times M_{at}(t)\big)\ \ /\ \ \big(R_{st}(t)\ +\ \varepsilon\big)\Big)\ \times\ \big(x_t{}^{d}(t)\ - x_s{}^{d}(t)\big) \tag{7}$$

$M_{at}\big(t\big):$ active gravitational mass of robot *t* at time *t*
$M_{ps}(t):$ passive gravitational mass of robot *s* at time *t*
$R_{st}(t):$ euclidean distance among two robots *s* and *t*
$\mathrm{G}\ (t):$ gravitational constant at time *t*
ε : a small constant
$x_t{}^{d}(t)$: position of robot *t* in direction *d*
$x_s{}^{d}(t):$ position of robot *s* in direction *d*

The total force acting on robot *s* is calculated as:

$$F_s^{\,d}(t) = \ \Sigma\, rand(t) \times F_{st}{}^{d}(t)\ , \quad \forall t = \ 1,\ 2,..,\ \mathrm{n},\ s \neq \ \mathrm{t} \tag{8}$$

$$F_s^{\,d}\big(t\big) = \ \Sigma rand\big(t\big) \times F_{st}{}^{d}\big(t\big)$$

Acceleration of robot *s* in direction *d* is equivalent to the force acting on robot *s* divided by the inertia mass of robot *s* and is updated by

$$a_s{}^{d}(t) \,=\, F_s^{\,d}(t)\ /M_{is}(t) \tag{9}$$

Robot velocity and the position at the next iteration are calculated by following equations

$$v_s^{\,d}(t) = \ rand \times v_s^{\,d}(t-1) + a_s^{\,d}(t-1) \tag{10}$$

$$x_s^{\,d}(t) = x_s^{\,d}(t-1) + v_s^{\,d}(t) \tag{11}$$

where $v_s^{\,d}(t), v_s^{\,d}(t-1)$: velocity of robot *s* in d direction at time *t* and *t*-1.

$a_s^{\,d}(t-1)$: acceleration of robot s in direction d at time *t*-1.
$rand$: random number which vary from 0 to 1.
$x_s^{\,d}(\mathrm{t})$, $x_s^{\,d}(t-1)$: position of robot *s* in direction *d* at time *t* and *t*-1.

Gravitational constant G is the function of G_0, α and iteration. Value of gravitational constant is decreased gradually to control the search accuracy. To enhance the performance of the proposed algorithm, the value of gravitational constant is calculated by the following equation:

$$\mathrm{G(t)} \ = \ \mathrm{G}_0 \times \exp\left(-\alpha \ \times \ \left(fit_s(t) + \ t\right)/\,T\right) \tag{12}$$

where G_0 and α are constants, t represents the current iteration number and *T* is the total number of iterations. $fit_s(t)$ is the fitness of robot *s* at time *t*. Value of gravitational constant is controlled to ensure the balanced exploration and exploitation.

Algorithm 4: GSA based Search

```
1: while (odor source is not find)
2: Evaluate the fitness of each robot.
3: Determine the best and worst fitness by following equations:
$gbest(t) = \max\left(fit_s(t)\right) \quad \forall s \ \ 1,\ 2,\ldots,\mathrm{N}$
Worst = 0
$fit_s(t)$:fitness value (odor concentration) of the robot s at
iteration t
4: Calculate the value of gravitational constant by equation
12.
5: Calculate the value of mass, force and acceleration of each
```

```
robot by equation 6, 8 and 9 respectively.
6: Robot's velocity and the position at the next iteration ($t$ +
1) is updated by the equations 10 and 11.
7: end while
```

Gravitational Particle Swarm Optimization Based Search

A new hybrid algorithm Gravitational Particle Swarm Optimization is proposed by combining the GSA and PSO. The main idea is to utilize the strength of both GSA and PSO to improve the search efficiency of the multi-robot system. In order to exploit the advantages of both the well-known algorithms, the proposed algorithm combines the social and cognitive component of PSO are combined with the acceleration (computed on the basis of attraction force applied by robots on each other) of GSA.

Having the cognitive component helps to have the knowledge about previous best location which is being utilized for future decision making. However keeping cognitive component increases the memory requirement and computational overhead (processing).

After every iteration, performance of each group is measured by the improvement in the fitness. If the fitness of group does not improve for a definite number of iterations (search counter), the position of best performing robot is updated by the following equation:

$$x_{best}(t) = x_{best}(t) + rand \tag{13}$$

where rand is a random number which varies from 0 to 10.

Velocity and position of each robot is updated by the following equations:

$$v_k(t+1) = \mathrm{w} * v_k\ (t) + \mathrm{c}_1 * rand_1 * (pbest_k - x_k(t)) + \mathrm{c}_2 * rand_2 * (gbest - x_k(t)) + \mathrm{c}_3 * a_k(t) \tag{14}$$

$$x_k(t+1) = x_k(t) + v_k(t+1) \tag{15}$$

where $v_k(t)$, $v_k(t+1)$ are the velocities of robot k at time t and t +1. w is the inertia weight. c_1, c_2 and c_3 are positive constants. rand_1 and rand_2 are random numbers varies from 0 to 1. $pbest_k$ is previous best position of robot k, *gbest* is the position of best performing robot. $a_k(t)$ is the acceleration of robot k at time t. $x_k(t)$, $x_k(t+1)$ are the positions of robot k at time t and t +1.

Algorithm 5: Gravitational Particle Swarm Optimization based Search

```
while (odor source is not find)
    Evaluate the fitness of each robot.
    Determine the gbest, pbest and worst by following equations:
    $gbest(t) = \max\left(fit_j(t)\right), \forall j\ 1,2,..,\mathrm{N}$
    $pbest(t) = \max\left(fit_j(t), fit_j(t-1)\right)$
    $\mathrm{worst} = 0$
    $fit_j\left(t\right)$ :fitness value (odor concentration) of the $j^{th}$ robot at iteration $t$
    if (fitness of group is not improved for a predefined number of iterations)
        Position of best performing robot is computed by 13
    end if
    Calculate the value of gravitational constant by equation 12.
    Calculate the value of mass, force and acceleration of each robot by equations 6, 8 and 9 respectively.
    Robot’s velocity and the position at the next iteration (t + 1) is updated by equations 14 and 15.
end while
```

Concatenated GPSO and GWO Based Search

A new algorithm is proposed by combining the GSA, GWO and PSO, with aim to further improve the search efficiency. Both of the possible combinations of GPSO and GWO such as, GPSO first then GWO second and GWO first and GPSO are applied to evaluate the performance and named as GPSO-GWO and GWO-GPSO.

GWO gets inspiration from grey wolves social hierarchy and hunting mechanism. To apply GWO, robots area considered as wolves and fitness of each robot is considered as solution. As population size is small, next move of robots is calculated by considering only alpha and beta. Fittest solution is regarded as alpha and second fittest solution as beta. Encircling behavior of prey is modeled by equations 16 and 17.

$$D = \left|C * X_{(pr)}\left(t\right) - X\left(t\right)\right| \tag{16}$$

$$X\left(t+1\right) = \left|X_{(pr)}\left(t\right) - A * D\right| \tag{17}$$

$R_{(pr)}$(t) is prey position vector at iteration t. R(t) is grey wolf position vector at iteration t. A and C are coefficient vectors.

$$A = 2 * a * \mathrm{rand}_{(1)} - a \tag{18}$$

$$C = 2 * \mathrm{rand}_{(2)} \tag{19}$$

where a, decreases linearly over course of iteration from 2 to 0. $\mathrm{rand}_{(1)}$ and $\mathrm{rand}_{(2)}$ are random number vectors between 0 and 1.

Alpha and beta are evaluated by calculating the fitness of all the robots. After finding alpha beta and delta equations from 20 to 23 are calculated and final position of remaining robots is updated by equation 24.

$$D_{(\mathrm{alpha})} = \left|C * X_{(alpha)}\left(t\right) - X\left(t\right)\right| \tag{20}$$

$$D_{(\mathrm{beta})} = \left|C * X_{(beta)}\left(t\right) - X\left(t\right)\right| \tag{21}$$

$$X_1\left(t+1\right) = \left|X_{(alpha)}\left(t\right) - A_1 * D_{(alpha)}\right| \tag{22}$$

$$X_2\left(t+1\right) = \left|X_{(beta)}\left(t\right) - A_1 * D_{(beta)}\right| \tag{23}$$

$$X\left(t+1\right) = X_1 + X_2 / 2 \tag{24}$$

Algorithm 6: GPSO-GWO based Search

```
while odor source is not find do
    Evaluate the Fitness of each robot.
    Determine the gbest, pbest and worst by following equations:
    $gbest(t) = \max(fit_j(t)), \forall j\ 1,2,..,N$
    $pbest(t) = \max(fit_j(t),\ fit_j(t-1))$
    worst = constant
    $fit_j(t)$: fitness value of the $j^{th}$ robot at iteration $t$

    if (fitness of group is not improved for a predefined number of iterations)
        Position of best performing robot is computed by 13
    end if
    Calculate the value of gravitational constant by equation 12.
    Calculate the value of mass, force and acceleration of each robot by equations 6, 8 and 9 respectively.
    Robot's velocity and the position at the next iteration (t + 1) is updated by equations 14 and 15.
    These updated positions are fed into the GWO as initial solutions.
    Evaluate alpha and beta on the basis of fitness.
    Evaluate equations from 20 to 23.
    Final positions of robots are calculated by equation 24.
end while
```

Algorithm 7: GWO-GPSO based Search

```
while odor source is not find do
    Evaluate the Fitness of each robot.
    Evaluate alpha and beta on the basis of fitness.
    Evaluate equations from 20 to 23.
    Evaluate positions of robots equation 24.
    These updated positions are fed into the GPSO as initial solutions.
    Determine the gbest, pbest and worst by following equations:
    $gbest(t) = \max(fit_j(t)), \forall j\ 1,2,..,N$
    $pbest(t) = \max(fit_j(t),\ fit_j(t-1))$
    worst = constant
```

```
$fit_j(t)$: fitness value of the $j^{th}$ robot at iteration $t$
8: if (fitness of group is not improved for a predefined number of iterations)
    Position of best performing robot is computed by 13
end if
9: Calculate the value of gravitational constant by equation 12.
10: Calculate the value of mass, force and acceleration of each robot by equations 6, 8 and 9 respectively.
11: Robot’s velocity and the position at the next iteration (t + 1) is updated by equations 14 and 15.
12: end while
```

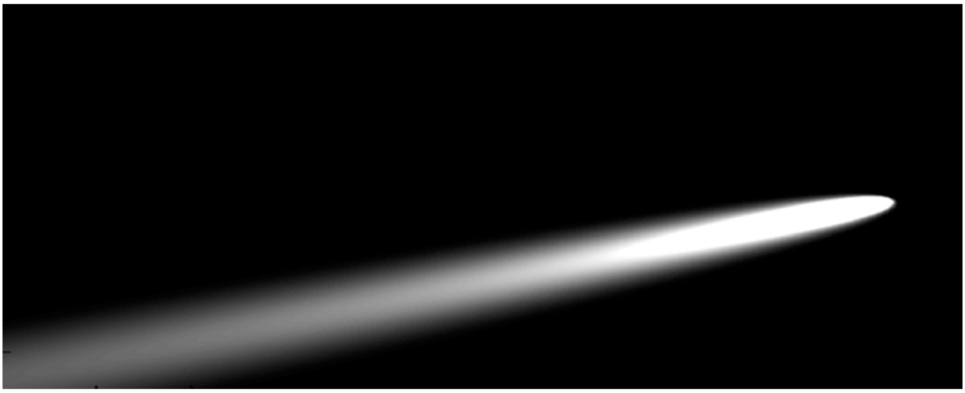
Odor Source Model

To model the odor distribution in the environment gaussian plume model is adapted from J. O. Hinze (Hinze, J. O. 1959) and Ishida (Ishida, H et al.1988). Figure 1 shows the odor source model. The Gaussian gas distribution is given by:

$$C(x,y) = \ \mathrm{q} \,/ \left(2 * \mu * \mathrm{k} * d_s \right) \times \exp\left((- \mathrm{U} \,/\, 2\mathrm{k}) \ \times \ (d_s - \Delta x)\right) \tag{25}$$

$$d_s = \ \surd\left(\mathrm{x}_\mathrm{m} - x\right)^2 + \left(\mathrm{y}_\mathrm{m} - y\right)^2 \tag{26}$$

Figure 1. Odor source model



$$\Delta x = \left(\mathrm{x}_{\mathrm{m}} - x\right) \ * \ \cos\theta \ + \ \left(\mathrm{y}_{\mathrm{m}} - y\right) \ * \ \sin\theta \tag{27}$$

where $\left(\mathrm{x}_{\mathrm{m}},\ \mathrm{y}_{\mathrm{m}}\right)$: Position of gas source

$\left(x, y\right)$: Position where gas density is measured
C : Plume concentration (ppm)
q : Gas emission rate (ml/s)
U : Wind speed (m/s)
k : Turbulent diffusion coefficient (m^2/s)
θ : Angle between x axis and upside wind direction
μ : Variance

SIMULATION SETUP

The proposed methods have been simulated in matlab. As mentioned earlier a rectangular area without any obstacles is considered for search space to compare the performance of proposed approaches GSA based search and Gravitational Particle Swarm Optimization based and Concatenated GPSO-GWO and GWO-GPSO. Environment dimension is set to 300* 300 pixels. Robot size is assumed to be same as pixel size of image. There is a single odor source in the environment with a constant release rate. Wind velocity is also kept constant at 1 m/s. Gaussian plume model is adopted to model the odor plume distribution in the environment which is suitable for indoor applications having stable wind flow. Robots are assumed to have perfect knowledge about the positions of other robot and are assumed to communicate perfectly without noise or channel interruptions.

Table 1 and Table 2 show the value of odor source model parameters and algorithm parameters.

Table 1. Odor Source Model Parameter

Parameter	Value
Release rate	200 g/s
μ	0.01
Θ	45
Wind Speed	1 m/s

Table 2. Algorithm Parameter

Parameter	Value
Number of Robots	6
W	0.5
c_1	1.6
c_2	0.6
c_3	0.6
G_0	50
α	5

SIMULATION RESULTS

GPSO, GSA, GPSO-GWO, GWO-GPSO and PSO based methods were run 20 times to calculate the average time required to finish the search, viz. localize odor source. The performance of algorithms is evaluated on the basis of the time required and the number of iterations required to finish the search.

Figure 2 shows the graph for time required to reach the maximum vicinity area versus the fitness of the best performing robot (among all 6 robots) during that time. Figure 3 shows the graph for the number of iterations required to reach the maximum vicinity area versus the fitness of best performing robot (among all 6 robots) during that iteration.

Figure 2. Performance comparison based on the time required to find the maximum intensity area

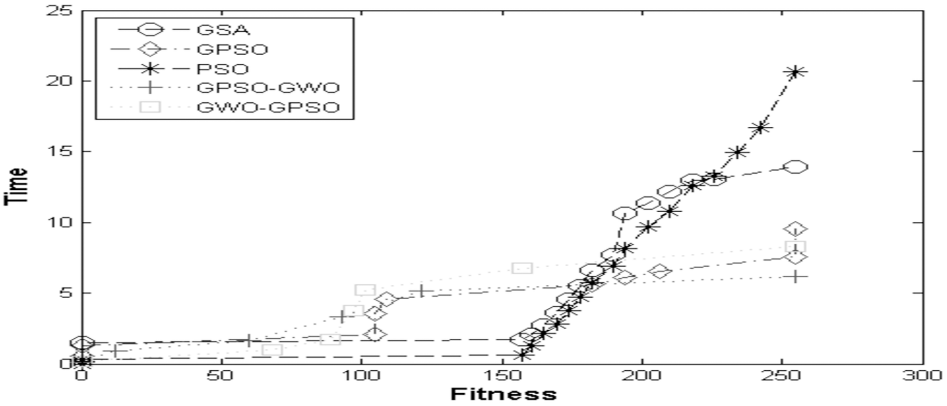
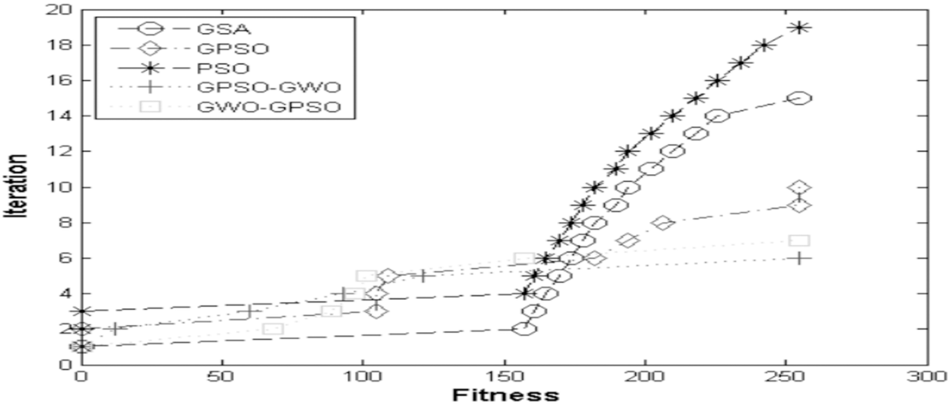


Figure 3. Performance comparison based on the iterations required to find the maximum intensity area



From figure 2 and 3 it could be observed that when search starts fitness value for all three methods increases slowly almost at the same speed. Performance of all the approaches is almost similar because the robots in the low intensity plume area having almost same fitness. As the search progresses and robots move to the higher intensity plume area fitness of robots increases, as the result GPSO, GPSO-GWO, GWO-GPSO and GSA based approach converges faster as compared to PSO based approach. GSA based approach converges faster because in GSA based method each robot's next move is governed by the force applied by all other robots. GPSO based method converges faster as compared to both GSA based search and PSO based search. This happens due to the reason that each robot's next move is governed by the force applied by all other robots, robot's previous best position and the position of best performing robot. Performance of GWO-GPSO is almost similar to GPSO-GWO based approach. GPSO-GWO performs best among all due to the reason it better manages the balance between exploration and exploitation during evaluation of next move of robots. Table 3 shows the average time and number of iterations required to finish the search.

For performance evaluation accuracy or precision of locating the odor source is also important, but focus is not given to the accuracy of source declaration. As mentioned earlier method deals with the first two phases of odor source localization i. e. Plume finding and plume traversal.

Table 3. Time and iterations required to locate the odor source

Method	Time required (sec)	Iterations
GPSO	10.44	11
GPSO-GWO	6.14	6
GWO-GPSO	8.1971	7
GSA	13.25	15
PSO	22.45	19

CONCLUSION AND FUTURE WORK

Current research in odor source localization is focusing on the problem of tracking the plume, but in order to track the plume it is necessary to find the plume. Problem of finding the plume and tracking the odor plume to localize the odor source is considered.

Three new methods, GSA based search, Concatenated GPSO and GWO and GPSO based approach are proposed for odor source localization. Odor source is modeled by Gaussian plume distribution model. The problem of local optima is handled by providing the concept of search counter. Proposed approaches is implemented in matlab environment and compared with the PSO based approach. Results shows that proposed methods perform better than PSO based search.

Field of multi robot odor source localization is an active area of research. In this work, odor source localization is done in simple environment which contains only single odor source. Future work involves finding odor source in complex environments with multiple odor sources.

REFERENCES

Awadalla, M., Lu, T. F., Tian, Z. F., Dally, B., & Liu, Z. (2013). 3D framework combining CFD and MATLAB techniques for plume source localization research. *Building and Environment*, *70*, 10–19. doi:10.1016/j.buildenv.2013.07.021

Couceiro, M. S., Rocha, R. P., & Ferreira, N. M. (2011, November). A novel multi-robot exploration approach based on particle swarm optimization algorithms. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics* (pp. 327-332). IEEE. 10.1109/SSRR.2011.6106751

Derr, K., & Manic, M. (2009, May). Multi-robot, multi-target particle swarm optimization search in noisy wireless environments. In *2009 2nd Conference on Human System Interactions* (pp. 81-86). IEEE. 10.1109/HSI.2009.5090958

Doctor, S., Venayagamoorthy, G. K., & Gudise, V. G. (2004, June). Optimal PSO for collective robotic search applications. In Evolutionary Computation, 2004. CEC2004. Congress on (Vol. 2, pp. 1390-1395). IEEE. doi:10.1109/CEC.2004.1331059

Eberhart, R. C., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science* (*Vol. 1*, pp. 39-43). Academic Press. 10.1109/MHS.1995.494215

Ferri, G., Caselli, E., Mattoli, V., Mondini, A., Mazzolai, B., & Dario, P. (2009). SPIRAL: A novel biologically-inspired algorithm for gas/odor source localization in an indoor environment with no strong airflow. *Robotics and Autonomous Systems*, *57*(4), 393–402. doi:10.1016/j.robot.2008.07.004

Gong, D. W., Qi, C. L., Zhang, Y., & Li, M. (2011, June). Modified particle swarm optimization for odor source localization of multi-robot. In *2011 IEEE Congress of Evolutionary Computation (CEC)* (pp. 130-136). IEEE. 10.1109/CEC.2011.5949609

Hayes, A. T., Martinoli, A., & Goodman, R. M. (2002). Distributed odor source localization. *IEEE Sensors Journal*, *2*(3), 260–271. doi:10.1109/JSEN.2002.800682

Hayes, A. T., Martinoli, A., & Goodman, R. M. (2003). Swarm robotic odor localization: Off-line optimization and validation with real robots. *Robotica*, *21*(04), 427–441. doi:10.1017/S0263574703004946

Hinze, J. O. (1959). Turbulence, 1975. New York: Academic Press.

Holland, O., & Melhuish, C. (1996). Some adaptive movements of animats with single symmetrical sensors. *From Animals to Animats, 4*(6), 55-64.

Ishida, H., Kagawa, Y., Nakamoto, T., & Moriizumi, T. (1996). Odor-source localization in the clean room by an autonomous mobile sensing system. *Sensors and Actuators. B, Chemical*, *33*(1-3), 115–121. doi:10.1016/0925-4005(96)01907-7

Ishida, H., Nakamoto, T., & Moriizumi, T. (1998). Remote sensing of gas/odor source location and concentration distribution using mobile system. *Sensors and Actuators. B, Chemical*, *49*(1), 52–57. doi:10.1016/S0925-4005(98)00036-7

Jatmiko, W., Jovan, F., Dhiemas, R. Y. S., Sakti, A. M., Ivan, F. M., Fukuda, T., & Sekiyama, K. (2011). Robots implementation for odor source localization using PSO algorithm. *WSEAS Transactions on Circuits and Systems*, *10*(4), 115–125.

Jatmiko, W., Nugraha, A., Effendi, R., Pambuko, W., Mardian, R., Sekiyama, K., & Fukuda, T. (2009). Localizing multiple odor sources in a dynamic environment based on modified niche particle swarm optimization with flow of wind. *WSEAS Transactions on Systems*, *8*(11), 1187–1196.

Jatmiko, W., Pambuko, W., Mursanto, P., Muis, A., Kusumoputro, B., Sekiyama, K., & Fukuda, T. (2009, November). Localizing multiple odor sources in dynamic environment using ranged subgroup PSO with flow of wind based on open dynamic engine library. In *Micro-NanoMechatronics and Human Science, 2009. MHS 2009. International Symposium on* (pp. 602-607). IEEE. 10.1109/MHS.2009.5351761

Kazadi, S., Goodman, R., Tsikata, D., Green, D., & Lin, H. (2000). An autonomous water vapor plume tracking robot using passive resistive polymer sensors. *Autonomous Robots*, *9*(2), 175–188. doi:10.1023/A:1008970418316

Kowadlo, G., & Russell, R. A. (2006). Using naïve physics for odor localization in a cluttered indoor environment. *Autonomous Robots*, *20*(3), 215–230. doi:10.100710514-006-7102-3

Kuwana, Y., & Shimoyama, I. (1998). A pheromone-guided mobile robot that behaves like a silkworm moth with living antennae as pheromone sensors. *The International Journal of Robotics Research*, *17*(9), 924–933. doi:10.1177/027836499801700902

Li, J. C., Meng, Q. H., & Liang, Q. (2007). Simulation study on robot active olfaction based on evolutionary gradient search. *Robot*, *29*, 234–238.

Li, J. G., Yang, J., Zhou, J. Y., Liu, J., & Lu, G. D. (2015). *Mapping odour sources with a mobile robot in a time variant airflow*. Academic Press.

Lilienthal, A., Reimann, D., & Zell, A. (2003). Gas source tracing with a mobile robot using an adapted moth strategy. In *Autonome Mobile Systeme 2003* (pp. 150–160). Springer Berlin Heidelberg. doi:10.1007/978-3-642-18986-9_16

Liu, Y., & Nejat, G. (2013). Robotic urban search and rescue: A survey from the control perspective. *Journal of Intelligent & Robotic Systems*, *72*(2), 147–165. doi:10.100710846-013-9822-x

Marjovi, A., & Marques, L. (2011). Multi-robot olfactory search in structured environments. *Robotics and Autonomous Systems*, *59*(11), 867–881.

Marques, L., & De Almeida, A. T. (2000, April). Electronic nose-based odour source localization. In *Advanced Motion Control, 2000. Proceedings. 6th International Workshop on* (pp. 36-40). IEEE. 10.1109/AMC.2000.862824

Marques, L., Nunes, U., & de Almeida, A. (2002, September). Cooperative odour field exploration with genetic algorithms. In *Proc. 5th Portuguese Conf. on Automatic Control (CONTROLO 2002)* (pp. 138-143). Academic Press.

Marques, L., Nunes, U., & de Almeida, A. T. (2006). Particle swarm-based olfactory guided search. *Autonomous Robots*, *20*(3), 277–287. doi:10.100710514-006-7567-0

Meng, Q. H., Yang, W. X., Wang, Y., Li, F., & Zeng, M. (2012). Adapting an ant colony metaphor for multi-robot chemical plume tracing. *Sensors (Basel)*, *12*(4), 4737–4763. doi:10.3390120404737 PMID:22666056

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61. doi:10.1016/j.advengsoft.2013.12.007

Mucherino, A., Papajorgji, P. J., & Pardalos, P. M. (2009). *Data mining in agriculture* (Vol. 34). Springer Science & Business Media.

Pugh, J., & Martinoli, A. (2007, April). Inspiring and modeling multi-robot search with particle swarm optimization. In *2007 IEEE Swarm Intelligence Symposium* (pp. 332-339). IEEE. 10.1109/SIS.2007.367956

Ramya, K., Kumar, K. P., & Rao, V. S. (2012). A survey on target tracking techniques in wireless sensor networks. *International Journal of Computer Science and Engineering Survey*, *3*(4), 93–108. doi:10.5121/ijcses.2012.3408

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, *179*(13), 2232–2248. doi:10.1016/j.ins.2009.03.004

Russell, R. A. (2003). Chemical source location and the robomole project. *Proceedings Australian Conference on Robotics and Automation.*

Sandini, G., Lucarini, G., & Varoli, M. (1993, July). Gradient driven self-organizing systems. In *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on* (Vol. 1, pp. 429-432). IEEE. 10.1109/IROS.1993.583132

Vergassola, M., Villermaux, E., & Shraiman, B. I. (2007). 'Infotaxis' as a strategy for searching without gradients. *Nature*, *445*(7126), 406–409. doi:10.1038/nature05464 PMID:17251974

Willis, M. A. (2002). *Biologically-Inspired Search Algorithms for Locating Unseen Odor Sources.* Belangar. doi:10.21236/ADA402125

Zarzhitsky, D., Spears, D., Thayer, D., & Spears, W. (2004, April). Agent-based chemical plume tracing using fluid dynamics. In *International Workshop on Formal Approaches to Agent-Based Systems* (pp. 146-160). Springer Berlin Heidelberg. 10.1007/978-3-540-30960-4_10

Zhang, Y., Zhang, J., Hao, G., & Zhang, W. (2015, August). Localizing odor source with multi-robot based on hybrid particle swarm optimization. In *Natural Computation (ICNC), 2015 11th International Conference on* (pp. 902-906). IEEE. 10.1109/ICNC.2015.7378110

Zhang, Y. L., Ma, X. P., & Miao, Y. Z. (2013). A Virtual Physics-Based Approach to Chemical Source Localization Using Mobile Robots. *Applied Mechanics and Materials*, *263*, 674–679.

Zou, Y., & Luo, D. (2008). A modified ant colony algorithm used for multi-robot odor source localization. *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, 502-509.

Compilation of References

ABB Robotics. (n.d.). *Technical reference manual: RAPID instructions, functions and data types*. ABB Robotics.

Abbas, N., Andersson, J., Iftikhar, M. U., & Weyns, D. (2016, April). Rigorous architectural reasoning for self-adaptive software systems. In Software Architectures (QRASA), 2016 Qualitative Reasoning about (pp. 11-18). IEEE. doi:10.1109/QRASA.2016.9

Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight, T. F. Jr, ... Weiss, R. (2000). Amorphous computing. *Communications of the ACM*, *43*(5), 74–82. doi:10.1145/332833.332842

ACS - American Chemical Society News Service Weekly Press. (2015, June 17). *Toward nanorobots that swim through blood to deliver drugs.* Author.

Agarwal, R., Bohner, M., O'Regan, D., & Peterson, A. (2002). Dynamic equations on time scales: A survey. *Journal of Computational and Applied Mathematics*, *141*(1-2), 1–26. doi:10.1016/S0377-0427(01)00432-0

Aggraval, Darzi, & Yang. (2014). Medical Sciences part II, Robotics in Sugrery - Past, Present and Future. In *Encyclopedia of Life Support Systems*. Retrieved from www.eolss.net

Aghili, F. (2012). A Prediction and Motion-Planning Scheme for Visually Guided Robotic Capturing of Free-Floating Tumbling Objects With Uncertain Dynamics. *IEEE T Robot*, *28*(3), 634–649. doi:10.1109/TRO.2011.2179581

Ahlbrandt, C. D., & Morian, C. (2002). Partial differential equations on time scales. *Journal of Computational and Applied Mathematics*, *141*(1-2), 35–55. doi:10.1016/S0377-0427(01)00434-4

Ai, C., Li, F. H., & Zhang, K. (2017). Detecting isolate safe areas in wireless sensor monitoring systems. *Tsinghua Science and Technology*, *22*(4), 427–443. doi:10.23919/TST.2017.7986945

Ali, H. (2014). Bionic Exoskeleton: History, Development and the Future. *International Conference on Advances in Engineering & Technology*.

Allali, S., & Benchaiba, M. (2017). Safe route guidance of rescue robots and agents based on hazard areas dissemination. *2017 International Conference on Mathematics and Information Technology (ICMIT)*. doi:10.1109/mathit.2017.8259692

Allali, S., Benchaïba, M., Ouzzani, F., & Menouar, H. (2018). No-collision grid based broadcast scheme and ant colony system with victim lifetime window for navigating robot in first aid applications. *Ad Hoc Networks*, *68*, 85–93. doi:10.1016/j.adhoc.2017.10.006

Allali, S., Menouar, H., & Benchaïba, M. (2016). Grid architecture for lightweight WSN-based area monitoring and alerts dissemination. *2016 International Symposium on Networks, Computers and Communications (ISNCC)*. 10.1109/ISNCC.2016.7746110

Aminzadeh, R., & Fotouhi, R. (2014). Novel design of a precision planter for a robotic assistaint farmer. *Proc. of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE.*

Amoui, M., Salehie, M., Mirarab, S., & Tahvildari, L. (2008, March). Adaptive action selection in autonomic software using reinforcement learning. In *Autonomic and Autonomous Systems, 2008. ICAS 2008. Fourth International Conference on* (pp. 175-181). IEEE. 10.1109/ICAS.2008.35

Anderson, N. H., & Butzin, C. A. (1974). Performance = Motivation X Ability: An Integration-Theoretical Analysis. *Journal of Personality and Social Psychology*, *30*(5), 598–604. doi:10.1037/h0037447

Army.mil. (n.d.). *Army Explores Futuristic Uniform for SOCOM*. The Official Homepage of the United States Army.

Asakura, K., Takikawa, H., Oishi, J., & Watanabe, T. (2010). An ad-hoc unicursal protocol for stable and long-lived communication systems in disaster situations. *International Journal of Knowledge and Web Intelligence*, *1*(3/4), 154. doi:10.1504/IJKWI.2010.034185

Asbeck, A. T., Dyer, R. J., Larusson, A. F., & Walsh, C. J. (2013). Biologically-Inspired Soft Exosuit. *IEEE International Conference on Rehabilitation Robotics*. 10.1109/ICORR.2013.6650455

Au, A. T. C., Arthur, T. C., & Kirch, R. F. (2000). EMG-Based Prediction of Shoulder and Elbow Kinematics in Able-Bodied and Spinal Cord Injured Individuals. *IEEE Transactions on Rehabilitation Engineering*, *8*(4), 471–480. doi:10.1109/86.895950 PMID:11204038

Avanzini, G. B., Ceriani, N. M., Zanchettin, A. M., Rocco, P., & Bascetta, L. (2014). Safety control of industrial robots based on a distributed distance sensor. *IEEE Transactions on Control Systems Technology*, *22*(6), 2127–2140. doi:10.1109/TCST.2014.2300696

Awadalla, M., Lu, T. F., Tian, Z. F., Dally, B., & Liu, Z. (2013). 3D framework combining CFD and MATLAB techniques for plume source localization research. *Building and Environment*, *70*, 10–19. doi:10.1016/j.buildenv.2013.07.021

Balabanovic, M., & Shoham, Y. (1997). Fab: Content-Based, Collaborative Recommendation. *Communications of the ACM*, *40*(3), 66–72. doi:10.1145/245108.245124

Bandala, A. A., & Dadios, E. P. (2016). Dynamic aggregation method for target enclosure using smoothed particle hydrodynamics technique: An implementation in quadrotor unmanned aerial vehicles (QUAV) swarm. *Journal of Advanced Computational Intelligence*, *20*(1), 84–91. doi:10.20965/jaciii.2016.p0084

Beattie, E., McGill, N., Parrotta, N., & Vladimirov, N. (n.d.). *Titan: A Powered, Upper-Body Exoskeleton.* Retrieved from http://www.contact.astm.org/studentmember/images/2012_SParrotta_Paper.pdf

Bennaceur, A., McCormick, C., Galán, J. G., Perera, C., Smith, A., Zisman, A., & Nuseibeh, B. (2016, May). Feed me, feed me: An exemplar for engineering adaptive software. In *Proceedings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems* (pp. 89-95). ACM. 10.1145/2897053.2897071

Beysens, D. A., Forgacs, G., & Glazier, J. A. (2000). Cell sorting is analogous to phase ordering in fluids. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(17), 9467–9471. doi:10.1073/pnas.97.17.9467 PMID:10944216

Bianco, C. G. L. (2009). Evaluation of Generalized Force Derivatives by Means of a Recursive Newton–Euler Approach. *IEEE Transactions on Robotics*, *25*(4), 954–959. doi:10.1109/TRO.2009.2024787

Bohner, M., & Peterson, A. (2001). *Dynamic Equations on Time Scales: An Introduction with Applications*. Boston: Birkäuser. doi:10.1007/978-1-4612-0201-1

Bonabeau, E. (1997). From classical models of morphogenesis to agent-based models of pattern formation. *Artificial Life*, *3*(3), 191–211. doi:10.1162/artl.1997.3.3.191 PMID:9385734

Boudali, H., Crouzen, P., & Stoelinga, M. (2010). A rigorous, compositional, and extensible framework for dynamic fault tree analysis. *Dependable and Secure Computing. IEEE Transactions on*, *7*(2), 128–143.

Boud, D., Keogh, R., & Walker, D. (1985). *Reflection: Turning experience into learning*. Routledge.

Bourgine, P., & Lesne, A. (Eds.). (2011). *Morphogenesis: Origins of Patterns and Shapes*. Berlin: Springer. doi:10.1007/978-3-642-13174-5

Bouzit, M., Burdea, G., Popescu, G., & Boian, R. (2002, June). The Rutgers Master II - New Design Force-Feedback Glove. *IEEE/ASME Transactions on Mechatronics*, *7*(2), 256–263. doi:10.1109/TMECH.2002.1011262

Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.). (2000). *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*. Washington, DC: National Academy Press.

Brown, M., Tsagarakis, N., & Caldwell, D. G. (2003). Exoskeletons for human force augmentation *Industrial Robot. International Journal (Toronto, Ont.)*, *30*(6), 592–602.

BS EN ISO 13482:2014 (2014). Robots and robotic devices. Safety requirements for personal care robots. British Standards Institution.

Buckler, B. (1996). A learning process model to achieve continuous improvement. *The Learning Organization*, *3*(3), 31–39. doi:10.1108/09696479610119660

Burgar, C. G., Lum, P. S., Shor, P. C., & Machiel Van der Loos, H. F. (2000). Development of Robots for Rehabilitation Therapy: The Palo Alto VA/Stanford Experience. *Journal of Rehabilitation Research and Development*, *37*, 663–673. PMID:11321002

Cámara, J., Correia, P., de Lemos, R., Garlan, D., Gomes, P., Schmerl, B., & Ventura, R. (2016). Incorporating architecture-based self-adaptation into an adaptive industrial software system. *Journal of Systems and Software*, *122*, 507–523. doi:10.1016/j.jss.2015.09.021

Cámara, J., Peng, W., Garlan, D., & Schmerl, B. (2017) Reasoning about Sensing Uncertainty in Decision-Making for Self-Adaptation. *Proceedings of the 15th International Workshop on Foundations of Coordination Languages and Self-Adaptive Systems*.

Caron, G., Dame, A., & Marchand, E. (2014). Direct model based visual tracking and pose estimation using mutual information. *Image and Vision Computing*, *32*(1), 54–63. doi:10.1016/j.imavis.2013.10.007

Carrillo, J. A., Fornasier, M., Toscani, G., & Vecil, F. (2010). Particle, kinetic, and hydrodynamic models of swarming. In G. Naldi, L. Pareschi, & G. Toscani (Eds.), *Mathematical Modeling of Collective Behavior in Socio-economic and Life Sciences* (pp. 297–336). Boston: Birkhäuser. doi:10.1007/978-0-8176-4946-3_12

Casolo, F., Cinquemani, S., & Cocetta, M. (2008). On Active Lower Limb Exoskeletons Actuators. *5th International Symposium on Mechatronics and Its Applications*.

Cavallaro, E. E., Rosen, J., Perry, J. C., & Burns, S. (2006, November). Real-Time Myoprocessors for a Neural Controlled Powered Exoskeleton Arm. *IEEE Transactions on Biomedical Engineering*, *53*(11), 2387–2396. doi:10.1109/TBME.2006.880883 PMID:17073345

Ceccarelli, M. (1998). Mechanisms Schemes in Teaching: A Historical Overview. *Journal of Mechanical Design*, *120*(4), 533–541. doi:10.1115/1.2829311

Ceccarelli, M. (2001). A Historical Perspective of Robotics Towards the Future. *J. of Robotics and Mechatronics*, *13*(3), 299–313. doi:10.20965/jrm.2001.p0299

Cernuzzi, L., & Zambonelli, F. (2005, July). Dealing with adaptive multi-agent organizations in the gaia methodology. In *International Workshop on Agent-Oriented Software Engineering* (pp. 109-123). Springer.

Cernuzzi, L., & Zambonelli, F. (2011). Adaptive organizational changes in agent-oriented methodologies. *The Knowledge Engineering Review*, *26*(2), 175–190. doi:10.1017/S0269888911000014

Chaumette, F., & Hutchinson, S. (2006). Visual servo control. I. Basic approaches, IEEE Robotics &amp;amp;amp. *Automation Magazine*, *13*(4), 82–90. doi:10.1109/MRA.2006.250573

Chaumette, F., & Hutchinson, S. (2007). Visual servo control - Part II: Advanced approaches. *IEEE Robotics & Automation Magazine*, *14*(1), 109–118. doi:10.1109/MRA.2007.339609

Chen, Z., Li, Z., Li, J., Chen, J., & Liu, Y. (2011). Quasi Real-Time Evaluation System for Seismic Disaster Based on Internet of Things. *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing (iThings/CPSCom).* doi:10.1109/ithings/cpscom.2011.112

Chen, D., Mahmud, N., Walker, M., Feng, L., Lönn, H., & Papadopoulos, Y. (2013). Systems Modeling with EAST-ADL for Fault Tree Analysis through HiP-HOPS. In *Proceedings of the 4th International Federation of Automatic Control (IFAC) Workshop on Dependable Control of Discrete Systems (DCDS), York*, (pp. 91-96). IFAC. 10.3182/20130904-3-UK-4041.00043

Cheng, B. H., De Lemos, R., Giese, H., Inverardi, P., Magee, J., Andersson, J., & Serugendo, G. D. M. (2009). Software engineering for self-adaptive systems: A research roadmap. In *Software engineering for self-adaptive systems* (pp. 1–26). Springer Berlin Heidelberg. doi:10.1007/978-3-642-02161-9_1

Chen, J., Yang, Y., & Wei, L. (2010). Research on the Approach of Task Decomposition in Soccer Robot System. *2010 International Conference on Digital Manufacturing & Automation (ICDMA).* 10.1109/ICDMA.2010.33

Chen, S. Y. (2012). Kalman Filter for Robot Vision: A Survey. *Ieee T Ind Electron*, *59*(11), 4409–4420. doi:10.1109/TIE.2011.2162714

Christensen, H. (2004). *European Service Robotics. A white paper on the status and opportunities of European Service Robots*. EURON-IFR.

Chu, A., Kazerooni, H., & Zoss, A. (2005). On the Biomimetic Design of the Berkeley Lower Extremity Exoskeleton (BLEEX). *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 4345-4352. 10.1109/ROBOT.2005.1570789

Chuang, Y.-L., D'Orsogna, M. R., Marthaler, D., Bertozzi, A. L., & Chayes, L. S. (2007). State transitions and the continuum limit for a 2D interacting, self-propelled particle system. *Physica D. Nonlinear Phenomena*, *232*(1), 33–47. doi:10.1016/j.physd.2007.05.007

Cickovski, T. M., Huang, C., Chaturvedi, R., Glimm, T., Hentschel, H. G. E., Alber, M. S., ... Izaguirre, J. A. (2005). A framework for three-dimensional simulation of morphogenesis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *2*(4), 273–278. doi:10.1109/TCBB.2005.46 PMID:17044166

Cooke, J., & Zeeman, E. C. (1976). A clock and wavefront model for control of the number of repeated structures during animal morphogenesis. *Journal of Theoretical Biology*, *58*(2), 455–476. doi:10.1016/S0022-5193(76)80131-2 PMID:940335

Correll, N. (2007). *Coordination Schemes for Distributed Boundary Coverage with a Swarm of Miniature Robots: Synthesis, Analysis and Experimental Validation* (PhD thesis). Ecole Polytechnique Fédérale de Lausanne.

Correll, N., & Martinoli, A. (2006). System identification of self-organizing robotic swarms. In M. Gini, & R. Voyles (Eds.), *Proceedings of the 8th International Symposium on Distributed Autonomous Robotic Systems (DARS 2006)* (pp. 31–40). Heidelberg, Germany: Springer.

Couceiro, M. S., Rocha, R. P., & Ferreira, N. M. (2011, November). A novel multi-robot exploration approach based on particle swarm optimization algorithms. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics* (pp. 327-332). IEEE. 10.1109/SSRR.2011.6106751

Crouzen, P., Hermanns, H., & Zhang, L. (2008). On the minimisation of acyclic models. In *International Conference on Concurrency Theory* (pp. 295-309). Springer Berlin Heidelberg.

Daerden, F., & Lefeber, D. (2002). Pneumatic Artificial Muscles: Actuators for Robotics and Automation. *European Journal of Mechanical and Environmental Engineering*, *47*(1), 11–21.

DARPA. (2014). Retrieved from www.darpa.mil

Dastani, M., Van Riemsdijk, M. B., Hulstijn, J., Dignum, F., & Meyer, J. J. C. (2004, July). Enacting and deacting roles in agent programming. In *International Workshop on Agent-Oriented Software Engineering* (pp. 189-204). Springer.

Dautenhahn, K. (2007). *Socially intelligent robots: Dimensions of human–robot interaction.* Philosophical Transactions. *Royal Society.*, *362*(1480), 679–704. PMID:17301026

de Gennes, P. G. (1992). Soft matter. *Science*, *256*(5056), 495–497. doi:10.1126cience.256.5056.495 PMID:17787946

Dehlinger, J., & Dugan, J. B. (2008). Analyzing dynamic fault trees derived from model-based system architectures. *Nuclear Engineering and Technology: An International Journal of the Korean Nuclear Society*, *40*(5), 365–374. doi:10.5516/NET.2008.40.5.365

Dellon, B., & Matsuoka, Y. (2007, March). Prosthetics, Exoskeletons, and Rehabilitation: Now and for the Future. *IEEE Robotics & Automation Magazine*. doi:10.1109/MRA.2007.339622

Dequéant, M. L., & Pourquié, O. (2008). Segmental patterning of the vertebrate embryonic axis. *Nature Reviews. Genetics*, *9*(5), 370–382. doi:10.1038/nrg2320 PMID:18414404

Derr, K., & Manic, M. (2009, May). Multi-robot, multi-target particle swarm optimization search in noisy wireless environments. In *2009 2nd Conference on Human System Interactions* (pp. 81-86). IEEE. 10.1109/HSI.2009.5090958

Doctor, S., Venayagamoorthy, G. K., & Gudise, V. G. (2004, June). Optimal PSO for collective robotic search applications. In Evolutionary Computation, 2004. CEC2004. Congress on (Vol. 2, pp. 1390-1395). IEEE. doi:10.1109/CEC.2004.1331059

Dollar, M. A., & Herr, H. (2008, February). Lower Extremity Exoskeletons and Active Orthoses: Challenges and State of the Art. *IEEE Transactions on Robotics*, *24*(1), 144–158. doi:10.1109/TRO.2008.915453

Dombrowksi, P., & Coval, D. (2012). Private First-Class Wing Man. *J Swanson Sch Eng, 4*.

Dong, M., Mao, X., Yin, J., Chang, Z., & Qi, Z. (2009). SADE: a development environment for adaptive multi-agent systems. *Principles of Practice in Multi-Agent Systems*, 516-524.

Dong, G. Q., & Zhu, Z. H. (2016). Autonomous robotic capture of non-cooperative target by adaptive extended Kalman filter based visual servo. *Acta Astronautica*, *122*, 209–218. doi:10.1016/j.actaastro.2016.02.003

Doursat, R. (2008). Organically grown architectures: Creating decentralized, autonomous systems by embryomorphic engineering. In R. P. Würtz (Ed.), *Organic Computing* (pp. 167–200). New York: Springer.

Doursat, R., Sayama, H., & Michel, O. (Eds.). (2012). *Morphogenetic Engineering: Toward Programmable Complex Systems*. New York: Springer. doi:10.1007/978-3-642-33902-8

Dugan, J. B., Bavuso, S. J., & Boyd, M. A. (1992). Dynamic fault-tree models for fault-tolerant computer systems. *Reliability*. *IEEE Transactions on*, *41*(3), 363–377.

Dutta, V., & Zielinska, T. (2015). Networking technologies for robotic applications. In *International Conference on Cyber Security (ICCS)*. California State University. Retrieved from http://iccs2015.iaasse.org/

EAST-ADL. (2010). *EAST-ADL Domain Model Specification, D4.1.1*. Retrieved from http://www.atesst.org/home/liblocal/docs/ATESST2_D4.1.1_EAST-ADL2-Specification_2010-06-02.pdf

Eberhart, R. C., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science* (*Vol. 1*, pp. 39-43). Academic Press. 10.1109/MHS.1995.494215

Edelman, G. M. (1988). *Topobiology: An Introduction to Molecular Embryology*. New York: Basic Books.

Engelberger, J. (1989). *Robotics in Service*. MIT Press. doi:10.1007/978-94-009-1099-7

Esfahani, N., & Malek, S. (2013). Uncertainty in self-adaptive software systems. In *Software Engineering for Self-Adaptive Systems II* (pp. 214–238). Springer Berlin Heidelberg. doi:10.1007/978-3-642-35813-5_9

Feddema, J. T., Lewis, C., & Schoenwald, D. A. (2002). Decentralized control of cooperative robotic vehicles: Theory and application. *IEEE Transactions on Robotics and Automation*, *18*(5), 852–864. doi:10.1109/TRA.2002.803466

Ferri, G., Caselli, E., Mattoli, V., Mondini, A., Mazzolai, B., & Dario, P. (2009). SPIRAL: A novel biologically-inspired algorithm for gas/odor source localization in an indoor environment with no strong airflow. *Robotics and Autonomous Systems*, *57*(4), 393–402. doi:10.1016/j.robot.2008.07.004

Ferris, D. P., & Lewis, C. L. (2009). Robotic Lower Limb Exoskeletons Using Proportional Myoelectric Control. *Conference Proceedings; ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*. PMID:19964579

Filieri, A., Maggio, M., Angelopoulos, K., D'ippolito, N., Gerostathopoulos, I., Hempel, A. B., ... Krikava, F. (2017). Control Strategies for Self-Adaptive Software Systems. *ACM Transactions on Autonomous and Adaptive Systems*, *11*(4), 24. doi:10.1145/3024188

Flacco, F., & De Luca, A. (2015). Discrete-time redundancy resolution at the velocity level with acceleration/torque optimization properties. *Robotics and Autonomous Systems*, *70*, 191–201. doi:10.1016/j.robot.2015.02.008

Fleischer, K. W. (1995). A multiple-mechanism developmental model for defining self-organizing geometric structures (Doctoral dissertation, California Institute of Technology, 1995). *Dissertation Abstracts International B*, 56/09, 4981, 1996.

Flores-Abad, A., Ma, O., Pham, K., & Ulrich, S. (2014). A review of space robotics technologies for on-orbit servicing. *Progress in Aerospace Sciences*, *68*, 1–26. doi:10.1016/j.paerosci.2014.03.002

Forgacs, G., & Newman, S. A. (2005). *Biological Physics of the Developing Embryo*. Cambridge, UK: Cambridge University Press. doi:10.1017/CBO9780511755576

Foundation for Spinal Cord Injury Prevention Care and Cure. (n.d.). Retrieved from http://www.fscip.org/facts.htm

Franklin House Marketing. (2017). *Canada Aurora Visitor Information.* Retrieved July 24, 2017, from http://www.canadaauroranetwork.com/

Fredericks, E. M. (2016, May). Automatically hardening a self-adaptive system against uncertainty. In *Proceedings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems* (pp. 16-27). ACM. 10.1145/2897053.2897059

Garlan, D., Cheng, S. W., Huang, A. C., Schmerl, B., & Steenkiste, P. (2004). Rainbow: Architecture-based self-adaptation with reusable infrastructure. *Computer*, *37*(10), 46–54. doi:10.1109/MC.2004.175

Garrec, P., Friconneau, J. P., Measson, Y., & Perrot, Y. (2008). ABLE, An Innovative Transparent Exoskeleton for the Upper-Limb. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1483-1488). IEEE. 10.1109/IROS.2008.4651012

Gasbarri, P., Sabatini, M., & Palmerini, G. B. (2014). Ground tests for vision based determination and control of formation flying spacecraft trajectories. *Acta Astronautica*, *102*, 378–391. doi:10.1016/j.actaastro.2013.11.035

Gazi, V., & Passino, K. M. (2003). Stability analysis of swarms. *IEEE Transactions on Automatic Control*, *48*(4), 692–697. doi:10.1109/TAC.2003.809765

Gelenbe, E., & Wu, F. (2012). Large scale simulation for human evacuation and rescue. *Computers & Mathematics with Applications (Oxford, England)*, *64*(12), 3869–3880. doi:10.1016/j.camwa.2012.03.056

Gerostathopoulos, I., Bures, T., Hnetynka, P., Keznikl, J., Kit, M., Plasil, F., & Plouzeau, N. (2016). Self-adaptation in software-intensive cyber–physical systems: From system goals to architecture configurations. *Journal of Systems and Software*, *122*, 378–397. doi:10.1016/j.jss.2016.02.028

Gil de la Iglesia, D., & Weyns, D. (2013, May). SA-MAS: Self-adaptation to enhance software qualities in multi-agent systems. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems* (pp. 1159-1160). International Foundation for Autonomous Agents and Multiagent Systems.

Gingold, R. A., & Monaghan, J. J. (1977). Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, *181*(3), 375–389. doi:10.1093/mnras/181.3.375

Goldstein, S. C., Campbell, J. D., & Mowry, T. C. (2005). Programmable matter. *Computer*, *38*(6), 99–101. doi:10.1109/MC.2005.198

Gong, D. W., Qi, C. L., Zhang, Y., & Li, M. (2011, June). Modified particle swarm optimization for odor source localization of multi-robot. In *2011 IEEE Congress of Evolutionary Computation (CEC)* (pp. 130-136). IEEE. 10.1109/CEC.2011.5949609

Gopura, R. A. R. C., & Kiguchi, K. (2009). Mechanical designs of Active Upper-Limb Exoskeleton Robots: State-of-the-art and Design Difficulties. *IEEE International Conference on Rehabilitation Robotics (ICORR)*, 178-187. 10.1109/ICORR.2009.5209630

Gopura, R. A. R. C., & Kiguchi, K. (2009). SUEFUL-7: A 7DOF Upper-Limb Exoskeleton Robot with Muscle-Model-Oriented EMG- Based Control. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 1126-1131. 10.1109/IROS.2009.5353935

Griffith, S., Subramanian, K., Scholz, J., Isbell, C. L., & Thomaz, A. (2013). Policy Shaping: Integrating Human Feedback with Reinforcement Learning. In *Proceedings of 27th Annual Conference on NIPS 2013* (pp. 2625-2633). La Jolla, CA: Neural Information Processing Systems Foundation, Inc.

Guizzo, E., & Goldstein, H. (2005). The Rise of the Body Bots [Robotic Exoskeletons]. Spectrum, IEEE, 42(10), 50-56.

Guo, S., Song, T., Xi, F. F., & Mohamed, R. P. (2017). Tip-Over Stability Analysis for a Wheeled Mobile Manipulator. ASME. J. Dyn. Sys. *Measurement and Control*, *139*(5), 054501–054501, 7. doi:10.1115/1.4035234

Habib, M. K. (2006). Mechatronics Engineering: The Evolution, the Needs and the Challenges. In *Proceedings of the 32nd Annual Conference of the IEEE Industrial Electronics Society, IECON 2006* (pp. 4510-4515). New York, NY: Institute of Electrical and Electronics Engineers (IEEE). 10.1109/IECON.2006.347925

Habib, M. K. (2007). Mechatronics: A Unifying Interdisciplinary and Intelligent Engineering Paradigm. *IEEE Industrial Electronics Magazine*, *1*(2), 12–24. doi:10.1109/MIE.2007.901480

Habib, M. K. (2008). *Humanitarian Demining: the Problem, Difficulties, Priorities, Demining Technology and the Challenge for Roboti*cs. *I-Tech Education and Publishing*.

Hamann, H. (2010). *Space-time Continuous Models of Swarm Robotic Systems: Supporting Global-to-Local Programming*. Berlin: Springer. doi:10.1007/978-3-642-13377-0

Hardesty, L. (2016). *Ingestible origami robot*. MIT News Office.

Harwin, W., Leiber, L., Austwick, G., & Dislis, C. (1998). Clinical Potential and Design of Programmable Mechanical Impedances for Orthotic Applications. *Robotica*, *16*(5), 523–530. doi:10.1017/S026357479800068X

Havlik, S. (2008). Land Robotic Vehicles for Demining. *I-Tech Education and Publishing*. doi:10.5772/5419

Hayes, A. T., Martinoli, A., & Goodman, R. M. (2002). Distributed odor source localization. *IEEE Sensors Journal*, *2*(3), 260–271. doi:10.1109/JSEN.2002.800682

Hayes, A. T., Martinoli, A., & Goodman, R. M. (2003). Swarm robotic odor localization: Off-line optimization and validation with real robots. *Robotica*, *21*(04), 427–441. doi:10.1017/S0263574703004946

Helbing, D., Schweitzer, F., Keltsch, J., & Molnar, P. (1997). Active walker model for the formation of human and animal trail systems. *Physical Review. E*, *56*(3), 2527–2539. doi:10.1103/PhysRevE.56.2527

Herr, H. (2009). Exoskeletons and Orthoses: Classification, Design Challenges, and Future Directions. *Journal of Neuroengineering and Rehabilitation*, *6*(21). PMID:19538735

Hilaire, V., Koukam, A., & Gruer, P. (2002, October). A mechanism for dynamic role playing. In *Net. ObjectDays: International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World* (pp. 36-48). Springer.

Hinze, J. O. (1959). Turbulence, 1975. New York: Academic Press.

Hogan, N., Krebs, H.I., Sharon, A., & Charnarong, J. (1995). *Interactive Robot Therapist*. Academic Press.

Hogg, T. (2006). Coordinating microscopic robots in viscous fluids. *Autonomous Agents and Multi-Agent Systems*, *14*(3), 271–305. doi:10.100710458-006-9004-3

Holland, O., & Melhuish, C. (1996). Some adaptive movements of animats with single symmetrical sensors. *From Animals to Animats, 4*(6), 55-64.

Housman, S. J., Le, V., Rahman, T., Sanchez, R. J., & Reinkensmeyer, D. J. (2007). Arm-Training with T-WREX After Chronic Stroke: Preliminary Results of a Randomized Controlled Trial. In *ICORR 2007. IEEE 10th International Conference on Rehabilitation Robotics* (pp. 562-568). IEEE. 10.1109/ICORR.2007.4428481

Huang, P. F., Wang, M., Meng, Z. J., Zhang, F., Liu, Z. X., & Chang, H. T. (2016). Reconfigurable spacecraft attitude takeover control in post-capture of target by space manipulators. *Journal of the Franklin Institute*, *353*(9), 1985–2008. doi:10.1016/j.jfranklin.2016.03.011

Huang, P. F., Zhang, F., Meng, Z. J., & Liu, Z. X. (2016). Adaptive control for space debris removal with uncertain kinematics, dynamics and states. *Acta Astronautica*, *128*, 416–430. doi:10.1016/j.actaastro.2016.07.043

Huang, P., Wang, D., Meng, Z., Zhang, F., & Guo, J. (2015). Adaptive Postcapture Backstepping Control for Tumbling Tethered Space Robot–Target Combination. *Journal of Guidance, Control, and Dynamics*, *39*, 1–7.

HULC®. (n.d.). Lockheed Martin.

IEEE Std.352. (1987). *IEEE Guide for General Principles of Reliability Analysis of Nuclear Power Generating Station Safety Systems.* ANSI/IEEE.

IFR Statistical Department. (2013). *Executive Summary.* World Robotics 2013: Service Robots.

IFR Statistical Department. (2014). *Executive Summary.* World Robotics 2014: Service Robots.

IFR Statistical Department. (2016). *Executive Summary.* World Robotics 2016: Service Robots.

Imran, M., Castillo, C., Diaz, F., & Vieweg, S. (2015). Processing Social Media Messages in Mass Emergency. *ACM Computing Surveys*, *47*(4), 1–38. doi:10.1145/2771588

Ishida, H., Kagawa, Y., Nakamoto, T., & Moriizumi, T. (1996). Odor-source localization in the clean room by an autonomous mobile sensing system. *Sensors and Actuators. B, Chemical*, *33*(1-3), 115–121. doi:10.1016/0925-4005(96)01907-7

Ishida, H., Nakamoto, T., & Moriizumi, T. (1998). Remote sensing of gas/odor source location and concentration distribution using mobile system. *Sensors and Actuators. B, Chemical*, *49*(1), 52–57. doi:10.1016/S0925-4005(98)00036-7

Ito, K., Tsuji, T., Kato, A., & Ito, M. (1992). EMG Pattern Classification for a Prosthetic Forearm with Three Degrees of Freedom. *IEEE International Workshop on Robot and Human Communication*, 69-74. 10.1109/ROMAN.1992.253907

Jain, S., Shah, R. C., Brunette, W., Borriello, G., & Roy, S. (2006). Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks. *Mobile Networks and Applications*, *11*(3), 327–339. doi:10.100711036-006-5186-9

Janabi-Sharifi, F., & Marey, M. (2010). A Kalman-Filter-Based Method for Pose Estimation in Visual Servoing. *Ieee T Robot*, *26*(5), 939–947. doi:10.1109/TRO.2010.2061290

Jankovic, M., Paul, J., & Kirchner, F. (2015). GNC architecture for autonomous robotic capture of a non-cooperative target: Preliminary concept design. *Advances in Space Research*, *57*(8), 1715–1736. doi:10.1016/j.asr.2015.05.018

Japan Meteorological Agency. (2017a). *Cherry flowering date (from 2011 to 2017 years).* Retrieved July 21, 2017, from http://www.data.jma.go.jp/sakura/data/sakura003_06.html

Japan Meteorological Agency. (2017b). *Search past weather data.* Retrieved July 25, 2017, from http://www.data.jma.go.jp/obd/stats/etrn/index.php

Jatmiko, W., Pambuko, W., Mursanto, P., Muis, A., Kusumoputro, B., Sekiyama, K., & Fukuda, T. (2009, November). Localizing multiple odor sources in dynamic environment using ranged subgroup PSO with flow of wind based on open dynamic engine library. In *Micro-NanoMechatronics and Human Science, 2009. MHS 2009. International Symposium on* (pp. 602-607). IEEE. 10.1109/MHS.2009.5351761

Jatmiko, W., Jovan, F., Dhiemas, R. Y. S., Sakti, A. M., Ivan, F. M., Fukuda, T., & Sekiyama, K. (2011). Robots implementation for odor source localization using PSO algorithm. *WSEAS Transactions on Circuits and Systems*, *10*(4), 115–125.

Jatmiko, W., Nugraha, A., Effendi, R., Pambuko, W., Mardian, R., Sekiyama, K., & Fukuda, T. (2009). Localizing multiple odor sources in a dynamic environment based on modified niche particle swarm optimization with flow of wind. *WSEAS Transactions on Systems*, *8*(11), 1187–1196.

Jensen, A. S., & Villadsen, J. (2013). A comparison of organization-centered and agent-centered multi-agent systems. *Artificial Intelligence Review*, *2*(3), 59.

Jiang, J. R., Lai, Y. L., & Deng, F. C. (2011). Mobile Robot Coordination and navigation with directional antennas in positionless Wireless Sensor Networks. *International Journal of Ad Hoc and Ubiquitous Computing*, *7*(4), 272. doi:10.1504/IJAHUC.2011.040775

Joshi, A., Vestal, S., & Binns, P. (2007). *Automatic generation of static fault trees from aadl models*. In Workshop on Architecting Dependable Systems of The 37th Annual IEEE/IFIP Int. Conference on Dependable Systems and Networks, Edinburgh, UK.

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*(1-2), 99–134. doi:10.1016/S0004-3702(98)00023-X

Kang, H., & Wang, J. (2013, July). Adaptive Control of 5 DOF Upper-Limb Exoskeleton Robot with Improved Safety. *ISA Transactions*, *52*(6), 844–852. doi:10.1016/j.isatra.2013.05.003 PMID:23906739

Kaplan, F., Oudeyer, P.-Y., Kubinyi, E., & Miklosi, A. (2002). Robotic clicker training. *Robotics and Autonomous Systems*, *38*(3-4), 197–206. doi:10.1016/S0921-8890(02)00168-9

Karbasian, M., Mehr, M. R., & Agharajabi, M. (2012). Improvement of the Reliability of Automatic Manufacture Systems by Using FTA Technique. *International Journal of Industrial Engineering*, *23*(2), 85–89.

Kazadi, S., Goodman, R., Tsikata, D., Green, D., & Lin, H. (2000). An autonomous water vapor plume tracking robot using passive resistive polymer sensors. *Autonomous Robots*, *9*(2), 175–188. doi:10.1023/A:1008970418316

Kazerooni, H., Steger, R., & Huang, L. (2006). Hybrid Control of the Berkeley Lower Extremity Exoskeleton (BLEEX). *The International Journal of Robotics Research*, 561-573.

Kazerooni, H. (2005). *Exoskeletons for Human Power Augmentation*. IEEE. doi:10.1109/IROS.2005.1545451

Kerman, M. C., Jiang, W., Blumberg, A. F., & Buttrey, S. E. (2009). *Event detection challenges, methods, and applications in natural and artificial systems*. Lockheed Martin MS2.

Kiguchi, K., & Hayashi, Y. (2012, August). An EMG-Based Control for an Upper-Limb Power-Assist Exoskeleton Robot. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, *42*(4), 1064–1071. doi:10.1109/TSMCB.2012.2185843 PMID:22334026

Kiguchi, K., Rahman, M. H., Sasaki, M., & Teramoto, K. (2008). Development of a 3DOF mobile exoskeleton robot for human upper-limb motion assist. *Robotics and Autonomous Systems*, *56*(8), 678–691. doi:10.1016/j.robot.2007.11.007

Kim, D., & Park, S. (2009, May). Reinforcement learning-based dynamic adaptation planning method for architecture-based self-managed software. In *Software Engineering for Adaptive and Self-Managing Systems* (pp. 76–85). IEEE.

Kirsh, R. F., & Au, A. T. C. (1997). EMG-Based Motion Intention Detection for Control of a Shoulder Neuroprosthesis. *IEEE International Conference of the Engineering in Medicine and Biology Society*, 5, 1944-1945. 10.1109/IEMBS.1997.758719

Kitano, H. (1996). Morphogenesis for evolvable systems. In E. Sanchez & M. Tomassini (Eds.), *Towards Evolvable Hardware: The Evolutionary Engineering Approach* (pp. 99–117). Berlin: Springer. doi:10.1007/3-540-61093-6_5

Ko, A., Lau, H. Y., & Lee, N. M. (2009). Ais based distributed wireless sensor network for mobile search and rescue robot tracking. *Journal of Mathematical Modelling and Algorithms*, *8*(2), 227–242. doi:10.100710852-009-9105-5

Kolagari, R. T., Chen, D., Lanusse, A., Librino, R., Lönn, H., Mahmud, N., ... Yakymets, N. (2015). Model-Based Analysis and Engineering of Automotive Architectures with EAST-ADL: Revisited. *International Journal of Conceptual Structures and Smart Applications*, *3*(2), 25–70. doi:10.4018/IJCSSA.2015070103

Komatsu, H., Endo, G., & Hodoshima, R. (2013). Basic Consideration About Optimal Control of a Quadruped Walking Robot During Slope Walking Motion. *IEEE Workshop on Advanced Robotics and Its Social Impacts (ARSO)*, 224–230. 10.1109/ARSO.2013.6705533

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms. *Reliability Engineering & System Safety*, *91*(9), 992–1007. doi:10.1016/j.ress.2005.11.018

Konda, T., Tensyo, S., & Yamaguchi, T. (2002a). LC-Learning: Phased Method for Average Reward Reinforcement Learning: Analysis of Optimal Criteria. In M. Ishizuka & A. Sattar (Eds.), *PRICAI2002: Trends in Artificial Intelligence, Lecture notes in Artificial Intelligence 2417* (pp. 198–207). Berlin, Germany: Springer. doi:10.1007/3-540-45683-X_23

Konda, T., Tensyo, S., & Yamaguchi, T. (2002b). Learning: Phased Method for Average Reward Reinforcement Learning (Preliminary Results). In M. Ishizuka & A. Sattar (Eds.), *PRICAI2002: Trends in Artificial Intelligence, Lecture notes in Artificial Intelligence 2417* (pp. 208–217). Berlin, Germany: Springer. doi:10.1007/3-540-45683-X_24

Konidaris, G., & Barto, A. (2006). Automonous Shaping: Knowledge Transfer in Reinforcement Learning. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 489-496). New York, NY: ACM

Konolige, K., Agrawal, M., & Sola, J. (2011). Large-scale visual odometry for rough terrain. In *Robotics Research* (pp. 201–212). Springer.

Kornienko, S., Kornienko, O., & Levi, P. (2005). Swarm embodiment – a new way for deriving emergent behavior in artificial swarms. In P. Levi, M. Schanz, R. Lafrenz, & V. Avrutin (Eds.), *Autonome Mobile Systeme* (pp. 25–32). Berlin: Springer.

Kowadlo, G., & Russell, R. A. (2006). Using naïve physics for odor localization in a cluttered indoor environment. *Autonomous Robots*, *20*(3), 215–230. doi:10.100710514-006-7102-3

Kravcik, M., Krogstie, B., Moore, A., Pammer, V., Pannese, L., Prilla, M., . . . Ullmann, T. (Eds.). (2013). In *Proceedings of the 3rd Workshop on Awareness and Reflection in Technology Enhanced Learning (ARTEL2013), in conjunction with the 8th European Conference on Technology Enhanced Learning(EC-TEL2013)*. Retrieved from http://ceur-ws.org/Vol-1103/

Krebs, H. I., Hogan, N., Aisen, M. L., Hogan, N., McDowell, F., & Volpe, B. T. (1997). The Effect of Robot-Assisted Therapy and Rehabilitative Training on Motor Recovery Following Stroke. [PubMed]. *Archives of Neurology*, *54*(4), 443–446. doi:10.1001/archneur.1997.00550160075019

Krogstie, B., & Prilla, M. (2012). Tool support for reflection in the workplace in the context of reflective learning cycles. In *Proceedings of the 2nd Workshop on Awareness and Reflection in Technology-Enhanced Learning (ARTEL 2012)* (pp. 57-72). Academic Press.

Krogstie, B., Prilla, M., & Pammer, V. (2013). Understanding and Supporting Reflective Learning Processes in the Workplace. In *Proceedings of the Eighth European Conference on Technology Enhanced Learning (EC-TEL 2013), LNCS* (vol. 8095, pp. 151-164). Springer.

Kurapati, S., Kolfschoten, G. L., Verbraeck, A., Drachsler, H., Specht, M., & Brazier, F. M. T. (2012). A Theoretical Framework for Shared Situational Awareness in Sociotechnical Systems. In *Proceedings of the 2nd Workshop on Awareness and Reflection in Technology-Enhanced Learning (ARTEL 2012)* (pp. 47-53). Academic Press.

Kuribayashi, K., Shimizu, S., Okimura, K., & Taniguchi, T. (1993). A Discrimination System Using Neural Networks for EMG-Control Prostheses-Integral type of EMG Signal Processing. *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1750-1755.

Kuwana, Y., & Shimoyama, I. (1998). A pheromone-guided mobile robot that behaves like a silkworm moth with living antennae as pheromone sensors. *The International Journal of Robotics Research*, *17*(9), 924–933. doi:10.1177/027836499801700902

Lalami, M. (2013). Output Feedback Control of an Actuated Lower Limb Orthosis with Bounded Input. *International Journal of Industrial Robots*, *40*(6).

Lalitharatne, T. D., Teramoto, K., Hayashi, Y., Nanayakkara, T., & Kiguchi, K. (2013). Evaluation of Fuzzy-Neuro Modifiers for Compensation of the Effects of Muscle Fatigue on EMG-Based Control to be Used in Upper-Limb Power-Assist Exoskeletons. *Journal of Advanced Mechanical Design, Systems and Manufacturing*, *7*(4), 736–751. doi:10.1299/jamdsm.7.736

Larouche, B. P., & Zhu, Z. H. (2014). Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control. *Autonomous Robots*, *37*(2), 157–167. doi:10.100710514-014-9383-2

Laschi, C. I., & Mazzolai, B. (2016). Lessons from Animals and Plants. IEEE Robotics and Automation, 23(3), 107-114.

Lee, S., Oh, J., & Lee, E. (2005, July). *An Architecture for Multi-agent Based Self-adaptive System in Mobile Environment*. IDEAL. doi:10.1007/11508069_64

Lerman, K., Martinoli, A., & Galstyan, A. (2005). A review of probabilistic macroscopic models for swarm robotic systems. In E. Şahin & W. M. Spears (Eds.), Swarm Robotics 2004 (pp. 143–152). Heidelberg, Germany: Springer. doi:10.1007/978-3-540-30552-1_12

Lerman, K. (2004). A model of adaptation in collaborative multi-agent systems. *Adaptive Behavior*, *12*(3–4), 187–198. doi:10.1177/105971230401200305

Lerman, K., & Galstyan, A. (2002). Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, *13*(2), 127–141. doi:10.1023/A:1019633424543

Li, J. G., Yang, J., Zhou, J. Y., Liu, J., & Lu, G. D. (2015). *Mapping odour sources with a mobile robot in a time variant airflow*. Academic Press.

Liang, B., Xu, Y. S., & Bergerman, M. (1998). Mapping a space manipulator to a dynamically equivalent manipulator. *J Dyn Syst-T Asme*, *120*(1), 1–7. doi:10.1115/1.2801316

Li, J. C., Meng, Q. H., & Liang, Q. (2007). Simulation study on robot active olfaction based on evolutionary gradient search. *Robot*, *29*, 234–238.

Lilienthal, A., Reimann, D., & Zell, A. (2003). Gas source tracing with a mobile robot using an adapted moth strategy. In *Autonome Mobile Systeme 2003* (pp. 150–160). Springer Berlin Heidelberg. doi:10.1007/978-3-642-18986-9_16

Li, N., Sun, M., Bi, Z., Su, Z., & Wang, C. (2014). A new methodology to support group decision-making for IoT-based emergency response systems. *Information Systems Frontiers*, *16*(5), 953–977. doi:10.100710796-013-9407-z

Lin, Y., Xi, F. F., Mohamed, R. P., & Tu, X. W. (2010). Calibration of Modular Reconfigurable Robots Based on a Hybrid Search Method. ASME. *Journal of Manufacturing Science and Engineering*, *132*(6), 061002–061002, 8. doi:10.1115/1.4002586

Litoiu, M., Shaw, M., Tamura, G., Villegas, N. M., Müller, H., Giese, H., & Rutten, E. (2017). What Can Control Theory Teach Us About Assurances in Self-Adaptive Software Systems? *Software Engineering for Self-Adaptive Systems 3: Assurances*, 9640.

Liu, J., Rojas-Cessa, R., & Dong, Z. (2016). Sensing, calculating, and disseminating evacuating routes during an indoor fire using a sensor and diffusion network. *2016 IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC).* doi:10.1109/icnsc.2016.7479014

Liu, G. R., & Liu, M. B. (2003). *Smoothed Particle Hydrodynamics: A Meshfree Particle Method.* Singapore: World Scientific. doi:10.1142/5340

Liu, J., Wang, Y., Li, B., & Ma, S. (2007). Current research, key performances and future development of search and rescue robots. Frontiers of Mechanical Engineering. *Frontiers of Mechanical Engineering*, *2*(4), 404–416. doi:10.100711465-007-0070-2

Liu, Y. G., & Liu, G. J. (2009). Modeling of tracked mobile manipulators with consideration of track–terrain and vehicle–manipulator interactions. *Robotics and Autonomous Systems*, *57*(11), 1065–1074. doi:10.1016/j.robot.2009.07.007

Liu, Y. G., & Liu, G. J. (2010). Interaction Analysis and Online Tip-Over Avoidance for a Reconfigurable Tracked Mobile Modular Manipulator Negotiating Slopes. *IEEE/ASME Transactions on Mechatronics*, *15*(4), 623–635. doi:10.1109/TMECH.2009.2031174

Liu, Y., & Nejat, G. (2013). Robotic urban search and rescue: A survey from the control perspective. *Journal of Intelligent & Robotic Systems*, *72*(2), 147–165. doi:10.100710846-013-9822-x

Lo, H. S., & Xie, S. Q. (2012). Exoskeleton robots for upper-limb rehabilitation: State of the art and future prospects. *Medical Engineering & Physics*, *34*(3), 261–268. doi:10.1016/j.medengphy.2011.10.004 PMID:22051085

Loukas, G., Timotheou, S., & Gelenbe, E. (2008). Robotic wireless network connection of civilians for emergency response operations. *2008 23rd International Symposium on Computer and Information Sciences (ISCIS).* doi:10.1109/iscis.2008.4717943

Lucas, L., DiCicco, M., & Matsuoka, Y. (2004). An EMG-Controlled Hand Exoskeleton for Natural Pinching. *Journal of Robotics and Mechatronics*, *16*(5), 482–488. doi:10.20965/jrm.2004.p0482

Lum, P. S., Burgar, C. G., Van der Loos, M., Shor, P. C., Majmundar, M., & Yap, R. (2006). MIME Robotic Device for Upper-Limb Neurorehabilitation in Subacute stroke subjects: A follow-up study. *Journal of Rehabilitation Research and Development*, *43*(5), 631. doi:10.1682/JRRD.2005.02.0044 PMID:17123204

MacLennan, B. J. (2010). Morphogenesis as a model for nano communication. *Nano Communication Networks*, *1*(3), 199–208. doi:10.1016/j.nancom.2010.09.007

MacLennan, B. J. (2011). Artificial morphogenesis as an example of embodied computation. *International Journal of Unconventional Computing*, *7*(1–2), 3–23.

MacLennan, B. J. (2012a). Embodied computation: Applying the physics of computation to artificial morphogenesis. *Parallel Processing Letters*, *22*(3), 1240013. doi:10.1142/S0129626412400130

MacLennan, B. J. (2012b). Molecular coordination of hierarchical self-assembly. *Nano Communication Networks*, *3*(2), 116–128. doi:10.1016/j.nancom.2012.01.004

MacLennan, B. J. (2014). Coordinating massive robot swarms. *International Journal of Robotics Applications and Technologies*, *2*(2), 1–19. doi:10.4018/IJRAT.2014070101

MacLennan, B. J. (in press). Coordinating swarms of microscopic agents to assemble complex structures. In Y. Tan (Ed.), *Swarm Intelligence — From Concepts to Applications (PBCE110)*. London: IET.

Mahadevan, S. (1996). Average Reward Reinforcement Learning: Foundations, Algorithms, and Empirical Results. *Machine Learning*, *22*(1-3), 159–195. doi:10.1007/BF00114727

Mahapatra, A., Roy, S. S., & Bhavanibhatla, K. (2015). Energy-Efficient Inverse Dynamic Model of a Hexapod Robot. *International Conference on Robotics, Automation, Control and Embedded Systems (RACE)*, 1–7. 10.1109/RACE.2015.7097237

Mahdavi-Hezavehi, S., Avgeriou, P., & Weyns, D. (2016). A classification framework of uncertainty in architecture-based Self-adaptive systems with multiple quality requirements. *Managing Trade-offs in Adaptable Software Architectures*, 45-78.

Mahdavi-Hezavehi, S., Durelli, V. H., Weyns, D., & Avgeriou, P. (2017). A systematic literature review on methods that handle multiple quality attributes in architecture-based self-adaptive systems. *Information and Software Technology*, *90*, 1-26.

Mahmud, N. (2012). *Dynamic Model-based Safety Analysis: From State Machines to Temporal Fault Trees* (Ph.D. dissertation). Department of Computer Science, University of Hull, Hull, UK.

Mahmud, N., & Mian, Z. (2013). Automatic generation of temporal fault trees from AADL models. In Safety, Reliability and Risk Analysis: Beyond the Horizon (pp. 2741–2749). CRC Press. doi:10.1201/b15938-415

Mahmud, N. (2015). Advanced fault tree synthesis for systems with dynamic aspects. In *Safety and Reliability of Complex Engineered Systems* (pp. 1635–1643). CRC Press. doi:10.1201/b19094-213

Mahmud, N. (2015). Improving Dependability of Robotics Systems, Experience from Application of Fault Tree Synthesis to Analysis of Transport Systems. *International Journal of Robotics Applications and Technologies*, *3*(2), 38–62. doi:10.4018/IJRAT.2015070103

Mahmud, N. (2017). A compositional symbolic calculus approach to producing reduced Markov chains. In *2017 Annual Reliability and Maintainability Symposium (RAMS)*. IEEE. 10.1109/RAM.2017.7889660

Mahmud, N., Papadopoulos, Y., & Walker, M. (2010). A Translation of State Machines to Temporal Fault Trees. In *Proceedings of the 40th IEEE/IFIP International Conference on Dependable Systems and Networks*, (pp. 45-51). IEEE. 10.1109/DSNW.2010.5542620

Mahmud, N., Walker, M., & Papadopoulos, Y. (2012). Compositional synthesis of temporal fault trees from state machines. *Performance Evaluation Review*, *39*(4), 79–88. doi:10.1145/2185395.2185444

Mao, X., Shan, L., Zhu, H., & Wang, J. (2008). An adaptive casteship mechanism for developing multi-agent systems. *International Journal of Computer Applications in Technology*, *31*(1-2), 17–34. doi:10.1504/IJCAT.2008.017716

Mao, Y., & Agrawal, S. K. (2012). Design of a Cable-Driven Arm Exoskeleton (CAREX) for Neural Rehabilitation. *IEEE Transactions on Robotics*, *28*(4), 922–931. doi:10.1109/TRO.2012.2189496

Marjovi, A., & Marques, L. (2011). Multi-robot olfactory search in structured environments. *Robotics and Autonomous Systems*, *59*(11), 867–881.

Marques, L., & De Almeida, A. T. (2000, April). Electronic nose-based odour source localization. In *Advanced Motion Control, 2000. Proceedings. 6th International Workshop on* (pp. 36-40). IEEE. 10.1109/AMC.2000.862824

Marques, L., Nunes, U., & de Almeida, A. (2002, September). Cooperative odour field exploration with genetic algorithms. In *Proc. 5th Portuguese Conf. on Automatic Control (CONTROLO 2002)* (pp. 138-143). Academic Press.

Marques, L., Nunes, U., & de Almeida, A. T. (2006). Particle swarm-based olfactory guided search. *Autonomous Robots*, *20*(3), 277–287. doi:10.100710514-006-7567-0

Marthi, B. (2007). Automatic shaping and decomposition of reward functions. In *Proceedings of the 24th international conference on Machine learning* (pp. 601-608). New York, NY: ACM. 10.1145/1273496.1273572

Martínez, C., Mondragón, I. F., Olivares-Méndez, M. A., & Campoy, P. (2011). On-board and ground visual pose estimation techniques for UAV control. *Journal of Intelligent & Robotic Systems*, *61*(1-4), 301–320. doi:10.100710846-010-9505-9

Martinoli, A. (1999). *Swarm Intelligence in Autonomous Collective Robotics: From Tools to the Analysis and Synthesis of Distributed Control Strategies* (PhD thesis). Ecole Polytechnique Fédérale de Lausanne.

Martinoli, A., Easton, K., & Agassounon, W. (2004). Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *The International Journal of Robotics Research*, *23*(4), 415–436. doi:10.1177/0278364904042197

Meinhardt, H. (1982). *Models of Biological Pattern Formation*. London: Academic Press.

Meng, Q. H., Yang, W. X., Wang, Y., Li, F., & Zeng, M. (2012). Adapting an ant colony metaphor for multi-robot chemical plume tracing. *Sensors (Basel)*, *12*(4), 4737–4763. doi:10.3390120404737 PMID:22666056

Meng, Y., & Jin, Y. (Eds.). (2011). *Bio-inspired Self-organizing Robotic Systems. In Studies in Computational Intelligence* (Vol. 355). Berlin: Springer. doi:10.1007/978-3-642-20760-0

Merle, G., Roussel, J. M., Lesage, J., & Bobbio, A. (2010). Probabilistic algebraic analysis of fault trees with priority dynamic gates and repeated events. *Reliability*. *IEEE Transactions on*, *59*(1), 250–261.

Mian, Z., Bottaci, L., Papadopoulos, Y., Sharvia, S., & Mahmud, N. (2015). Model Transformation for Multi-objective Architecture Optimisation of Dependable Systems. In *Dependability Problems of Complex Information Systems* (pp. 91–110). Springer. doi:10.1007/978-3-319-08964-5_6

Mihelj, M., Nef, T., & Riener, R. (2007). ARMin II-7 DoF Rehabilitation Robot: Mechanics and Kinematics. In *IEEE International Conference on Robotics and Automation* (pp. 4120-4125). IEEE. 10.1109/ROBOT.2007.364112

Milford, M., & George, A. (2013). Featureless Visual Processing for SLAM in Changing Outdoor Environments. *Springer Tracts in Advanced Robotics Field and Service Robotics*, 569-583. doi:10.1007/978-3-642-40686-7_38

Millán, J. D. R., Rupp, R., Müller-Putz, G. R., Murray-Smith, R., Giugliemma, C., Tangermann, M., ... Mattia, D. (2010). Combining Brain–Computer Interfaces and Assistive Technologies: State-of-the-Art and Challenges. *Frontiers in Neuroscience*, *4*, 161. PMID:20877434

Miller, C., Ruiz-Torres, R., Stienen, A. H. A., & Dewald, J. P. A. (2009). A Wrist and Finger Force Sensor Module for Use During Movements of the Upper Limb in Chronic Hemiparetic Stroke. *IEEE Transactions on Biomedical Engineering*, *56*(9), 2312–2317. doi:10.1109/TBME.2009.2026057 PMID:19567336

Mintchev, S., & Floreano, D. (2016). Adaptive Motphology. *IEEE Robotics and Automation*, *23*(3), 42–54. doi:10.1109/MRA.2016.2580593

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61. doi:10.1016/j.advengsoft.2013.12.007

Mizen, N. J. (1965). *Preliminary Design for the Shoulders and Arms of a Powered, Exoskeletal Structure.* Cornell Aeronautical Laboratory Report VO-1692-V-4.

Moore, A., Pammer, V., Pannese, L., Prilla, M., Rajagopal, K., Reinhardt, W., . . . Voigt, C. (Eds.). (2012). *Proceedings of the 2nd Workshop on Awareness and Reflection in Technology Enhanced Learning (ARTEL2012), in conjunction with the 8th European Conference on Technology Enhanced Learning(EC-TEL2012)*. Retrieved from http://ceur-ws.org/Vol-931/

Muccini, H., Sharaf, M., & Weyns, D. (2016). Self-adaptation for cyber-physical systems: a systematic literature review. In *Proceedings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems* (pp. 75-81). ACM. 10.1145/2897053.2897069

Mucherino, A., Papajorgji, P. J., & Pardalos, P. M. (2009). *Data mining in agriculture* (Vol. 34). Springer Science & Business Media.

Murata, S., & Kurokawa, H. (2007). Self-reconfigurable robots: Shape-changing cellular robots can exceed conventional robot flexibility. *IEEE Robotics & Automation Magazine*, *14*(1), 71–78. doi:10.1109/MRA.2007.339607

Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., & Erkmen, A. M. (2008). Search and Rescue Robotics. Springer Handbook of Robotics, 1151-1173. doi:10.1007/978-3-540-30301-5_51

Murray, L., Liu, W., Winfield, A., Timmis, J., & Tyrrell, A. (2012). Analysing the Reliability of a Self-reconfigurable Modular Robotic System. In Bio-Inspired Models of Networks, Information, and Computing Systems (pp. 44-58). Springer Berlin Heidelberg. doi:10.1007/978-3-642-32711-7_4

Nagatani, K., Otake, K., & Yoshida, K. (2013). Three-Dimensional Thermography Mapping for Mobile Rescue Robots. *Springer Tracts in Advanced Robotics Field and Service Robotics,* 49-63. doi:10.1007/978-3-642-40686-7_4

Nagpal, R., Kondacs, A., & Chang, C. (2003). Programming methodology for biologically-inspired self-assembling systems. *AAAI Spring Symposium on Computational Synthesis: From Basic Building Blocks to High Level Functionality*. Retrieved from http://www.aaai.org/Papers/Symposia/Spring/2003/SS-03-02/SS03-02-024.pdf

Nasridinov, A., Ihm, S., Jeong, Y., & Park, Y. (2014). Event Detection in Wireless Sensor Networks: Survey and Challenges. *Lecture Notes in Electrical Engineering Mobile, Ubiquitous, and Intelligent Computing (MUSIC),* 585-590. doi:10.1007/978-3-642-40675-1_87

Nef, T., Mihelj, M., & Riener, R. (2007). ARMin: A Robot for Patient-Cooperative Arm Therapy. *Medical & Biological Engineering & Computing*, *45*(9), 887–900. doi:10.100711517-007-0226-6 PMID:17674069

Nerurkar, E. D., Zhou, K. X., & Roumeliotis, S. I. (2011). A hybrid estimation framework for cooperative localization under communication constraints. *International Conference on Intelligent Robots and Systems 2011 (IEEE/RSJ).* 10.1109/IROS.2011.6094684

Ng, A. Y., Harada, D., & Russell, S. J. (1999). Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *Proceedings of the 16th International Conference on Machine Learning* (pp. 278-287). New York, NY: ACM

Ng, A. Y., & Russell, S. (2000). Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML2000)* (pp. 663-670). San Francisco, CA: Morgan Kaufmann Publishers Inc.

Nüsslein-Volhard, C. (2008). *Coming to Life: How Genes Drive Development*. Carlsbad, CA: Kales.

Odell, J. J., Parunak, H. V. D., Brueckner, S., & Sauter, J. (2003, July). Temporal aspects of dynamic role assignment. In *International Workshop on Agent-Oriented Software Engineering* (pp. 201-213). Springer.

Okada, Y., Nagatani, K., Yoshida, K., Tadokoro, S., Yoshida, T., & Koyanagi, E. (2011). Shared autonomy system for tracked vehicles on rough terrain based on continuous three-dimensional terrain scanning. *Journal of Field Robotics*, *28*(6), 875–893. doi:10.1002/rob.20416

Pac, M. R., Erkmen, A. M., & Erkmen, I. (2007). Control of Robotic Swarm Behaviors Based on Smoothed Particle Hydrodynamics. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4194–4200). IEEE Press. 10.1109/IROS.2007.4399437

Pandey, A., Moreno, G. A., Cámara, J., & Garlan, D. (2016, September). Hybrid planning for decision making in self-adaptive systems. In *Self-Adaptive and Self-Organizing Systems (SASO), 2016 IEEE 10th International Conference on* (pp. 130-139). IEEE. 10.1109/SASO.2016.19

Pedersen, S. M., Fountas, S., & Blackmore, S. (2008). *Agricultural Robots – Applications and Economic Perspectives*. Retrieved from www.intechopen.com

Perkinson, J. R., & Shafai, B. (2005). A decentralized control algorithm for scalable robotic swarms based on mesh-free particle hydrodynamics. In *Proceedings of the IASTED International Conference on Robotics and Applications* (pp. 102–107). Cambridge, MA: IASTED.

Perry, J. C., Rosen, J., & Burns, S. (2007). Upper-limb powered exoskeleton design. *Mechatronics, IEEE/ASME Transactions on, 12*(4), 408-417.

Perry, J., & Rosen, J. (2006). Design of a 7 degree-of-freedom upper-limb powered exoskeleton. *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, 805 –810. 10.1109/BIOROB.2006.1639189

Peterson, G. B. (2000). The Discovery of Shaping or B.F. Skinner's Big Surprise. *The Clicker Journal: The Magazine for Animal Trainers*, *43*, 6–13.

Peterson, G. B. (2001). The world's first look at shaping: B. F. Skinner's gutsy gamble. *The Clicker Journal: The Magazine for Animal Trainers*, *2001*, 14–21.

Pimenta, L. C. A., Mendes, M. L., Mesquita, R. C., & Pereira, G. A. S. (2007). Fluids in electrostatic fields: An analogy for multirobot control. *IEEE Transactions on Magnetics*, *43*(4), 1765–1768. doi:10.1109/TMAG.2007.892514

Pimenta, L. C. A., Pereira, G. A. S., Michael, N., Mesquita, R. C., Bosque, M. M., Chaimowicz, L., & Kumar, V. (2013). Swarm coordination based on smoothed particle hydrodynamics technique. *IEEE Transactions on Robotics*, *29*(2), 383–399. doi:10.1109/TRO.2012.2234294

Pneumatic-Hydraulic Actuator System. (n.d.). *Patent US 4647004A*. Washington, DC: US Patent Office.

Pons, J. L., Moreno, J. C., Brunetti, F. J., & Rocon, E. (2007). Lower-limb Wearable Exoskeleton. In *Rehabilitation Robotics* (pp. 772-474). InTech. Available from: http://www.intechopen.com/books/rehabilitation_robotics/lower-limb_wearable_exoskeelton, 2007.

Pons, J.L., Rocon, E., Ruiz, A.F., & Morenso, J.C. (2007). Upper-Limb Robotic Rehabilitation Exoskeleton: Tremor Suppression. *Rehabilitation Robotics*, 453-470.

Popescu, R., Staikopoulos, A., Brogi, A., Liu, P., & Clarke, S. (2012). A formalized, taxonomy-driven approach to cross-layer application adaptation. *ACM Transactions on Autonomous and Adaptive Systems*, *7*(1), 7. doi:10.1145/2168260.2168267

Pransky, J. (1996). Service robots – how should we define them? *Service Robot. International Journal (Toronto, Ont.)*, *2*(1), 4–5.

Pratt, J. E., Krupp, B. T., & Morse, C. J. (2014). The RoboKnee: An Exoskeleton for Enhancing Strength and Endurance During Walking. *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*.

Pryor, K. (2006). *Don't Shoot the Dog! The New Art of Teaching and Training*. Lydney, UK: Ringpress Books.

Pugh, J., & Martinoli, A. (2007, April). Inspiring and modeling multi-robot search with particle swarm optimization. In *2007 IEEE Swarm Intelligence Symposium* (pp. 332-339). IEEE. 10.1109/SIS.2007.367956

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY: John Wiley & Sons, Inc. doi:10.1002/9780470316887

Qu, W. W., Shi, X., Dong, H. Y., Feng, P. J., Zhu, L. S., & Ke, Y. L. (2014). Simulation and Test on Process of Percussive Impact Riveting. *Journal of Zhejiang University (Engineering Science)*, *48*(8), 1411–1418.

Ramya, K., Kumar, K. P., & Rao, V. S. (2012). A survey on target tracking techniques in wireless sensor networks. *International Journal of Computer Science and Engineering Survey*, *3*(4), 93–108. doi:10.5121/ijcses.2012.3408

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, *179*(13), 2232–2248. doi:10.1016/j.ins.2009.03.004

Rauzy, A. (2002). Mode automata and their compilation into fault trees. *Reliability Engineering & System Safety*, *78*(1), 1–12. doi:10.1016/S0951-8320(02)00042-X

Rauzy, A., Châtelet, E., Dutuit, Y., & Bérenguer, C. (2003). A practical comparison of methods to assess sum-of-products. *Reliability Engineering & System Safety*, *79*(1), 33–42. doi:10.1016/S0951-8320(02)00165-5

Raytheon, X. O. S. 2 Exoskeleton, Second-Generation Robotics Suit, United States of America. (n.d.). Retrieved from http://www.army-technology.com/projects/raytheon-xos-2-exoskeleton-us/

Rehman, B. U., Focchi, M., & Lee, J. (2016). Towards a multi-legged mobile manipulator. *Robotics and Automation (ICRA), IEEE International Conference on. IEEE*, 3618-3624. 10.1109/ICRA.2016.7487545

Reinhardt, W., & Christian, M. (2011). Awareness in Learning Networks. In *Proceedings of the PLE Conference 2011 (ARTEL2011)* (pp. 12-20). Academic Press.

Reinkensmeyer, D. J., Dewald, J. P. A., & Rymer, W. Z. (1999). Guidance-Based Quantification of Arm Impairment Following Brain Injury: A pilot study. *IEEE Transactions on Rehabilitation Engineering*, *7*(1), 1–11. doi:10.1109/86.750543 PMID:10188602

Ren, Y., Park, H., & Zhang, L. (2009). Developing a Whole-Arm Exoskeleton Robot with Hand Opening and Closing Mechanism for Upper Limb Stroke Rehabilitation. *IEEE 11th International Conference on Rehabilitation Robotics*.

ReWalk Robotics Announces Reimbursement Coverage by Major German Insurance Company. (n.d.). Retrieved from http://www.rewalk.com/rewalk-robotics-announces-reimbursement-coverage-by-major-german- insurance-company/

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *Computer Graphics*, *21*(4), 25–34. doi:10.1145/37402.37406

Riecken, D. (2000). Introduction: Personalized views of personalization. *Communications of the ACM*, *43*(8), 26–28. doi:10.1145/345124.345133

Rocon, E., Belda-Lois, J. M., Ruiz, A. F., Manto, M., Moreno, J. C., & Pons, J. L. (2007, September). Design and Validation of a Rehabilitation Robotic Exoskeleton for Tremor Assessment and Suppression. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *15*(3), 367–378. doi:10.1109/TNSRE.2007.903917 PMID:17894269

Roderick, S. N., & Carignan, C. R. (2005). An Approach to Designing Software Safety Systems for Rehabilitation Robots. In *Proceedings of the 2005 IEEE 9th International Conference on Rehabilitation Robotics* (pp. 252-257). IEEE. 10.1109/ICORR.2005.1501096

Rosen, J., Perry, J. C., Manning, N., Burns, S., & Hannaford, B. (2005). The Human Arm Kinematics and Dynamics During Daily Activities-Toward a 7 DOF Upper Limb Powered Exoskeleton. In *ICAR'05. Proceedings of 12th International Conference on Advanced Robotics* (pp. 532-539). IEEE. 10.1109/ICAR.2005.1507460

Rosheim, M. E. (1994). *Robot Evolution: The Development of Anthrobotics*. New York: John Wiley.

Rossiter, J., & Hauser, H. (2016). Soft Robotics - The Next Industrial Revolution? *IEEE Robotics and Automation*, *23*(3), 17–20. doi:10.1109/MRA.2016.2588018

Russell, R. A. (2003). Chemical source location and the robomole project. *Proceedings Australian Conference on Robotics and Automation*.

Sabatini, M., Palmerini, G. B., & Gasbarri, P. (2015). A testbed for visual based navigation and control during space rendezvous operations. *Acta Astronautica*, *117*, 184–196. doi:10.1016/j.actaastro.2015.07.026

Sabatucci, L., Seidita, V., & Cossentino, M. (2017, June). The Four Types of Self-adaptive Systems: A Metamodel. In *International Conference on Intelligent Interactive Multimedia Systems and Services* (pp. 440-450). Springer.

Salazar-Ciudad, I., Jernvall, J., & Newman, S. (2003). Mechanisms of pattern formation in development and evolution. *Development*, *130*(10), 2027–2037. doi:10.1242/dev.00425 PMID:12668618

Salehie, M., & Tahvildari, L. (2009). Self-adaptive software: Landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems, 4*(2), 14.

Sanchez, R. J., Liu, J., Rao, S., Shah, P., Smith, R., Rahman, T., & Reinkensmeyer, D. J. (2006). Automating Arm Movement Training Following Severe Stroke: Functional Exercises with Quantitative Feedback in a Gravity-Reduced Environment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *14*(3), 378–389. doi:10.1109/TNSRE.2006.881553 PMID:17009498

Sandini, G., Lucarini, G., & Varoli, M. (1993, July). Gradient driven self-organizing systems. In *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on* (Vol. 1, pp. 429-432). IEEE. 10.1109/IROS.1993.583132

Sansoni, S., Wodehouse, A., & Buis, A. (2014). *The Aesthetics of Prosthetic Design: From Theory to Practice*. International Design Conference – Design, Dubrovnik, Croatia.

Sarakoglou, I., Tsagarakis, N. G., & Caldwell, D. G. (2004). Occupational and Physical Therapy Using a Hand Exoskeleton Based Exerciser. *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Satoh, K., & Yamaguchi, T. (2006). Preparing various policies for interactive reinforcement learning. In *Proceedings of the SICE-ICASE International Joint Conference 2006 (SICE-ICASE 2006)* (pp. 2440-2444). New York, NY: Institute of Electrical and Electronics Engineers (IEEE). 10.1109/SICE.2006.315139

Sawicki, G. S., & Ferris, D. P. (2009, June). A Pneumatically Powered Knee-Ankle Foot Orthosis (KAFO) with Myoelectric Activation and Inhibition. *Journal of Neuroengineering and Rehabilitation*, *6*(1), 23. doi:10.1186/1743-0003-6-23 PMID:19549338

Schabowksy, C.N., Godfrey, S.B., Holley, R.J., & Lum, P.S. (2010). Development and Pilot Testing of HEXORR: Hand EXOskeleton Rehabilitation Robot. *Journal of NeuroEngineering and Rehabilitation, 7*(37).

Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-Commerce Recommendation Applications. *Journal of Data Mining and Knowledge Discovery*, *5*(1/2), 115–153. doi:10.1023/A:1009804230409

Schiele, A. (2007). Undesired Constraint Forces in Non-Ergonomic Wearable Exoskeletons. *Extended Abstract for IROS'07 Workshop on Assistive Technologies: Rehabilitation and Assistive Robotics*.

Schiele, A., & van der Helm, F. C. T. (2006, December). Kinematic Design to Improve Ergonomics in Human Machine Interaction. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *14*(4), 456–469. doi:10.1109/TNSRE.2006.881565 PMID:17190037

Schnieders, T. M., Stone, R. T., Oviatt, T., & Danford-Klein, E. (2017). ARCTiC LawE- An Upper Body Exoskeleton for Firearm Training. *Augmented Human Research*, *2*(1), 1. doi:10.100741133-017-0004-4

Schweitzer, F. (2002). Brownian agent models for swarm and chemotactic interaction. In D. Polani, J. Kim, T. Martinetz (Eds.), *Fifth German Workshop on Artificial Life: Abstracting and Synthesizing the Principles of Living Systems* (pp. 181–190). Berlin: Akademische Verlagsgesellschaft Aka.

Schweitzer, F. (2003). Brownian agents and active particles. In On the Emergence of Complex Behavior in the Natural and Social Sciences (pp. 295–333). Berlin: Springer.

Schweitzer, F., Lao, K., & Family, F. (1997). Active random walkers simulate trunk trail formation by ants. *Bio Systems*, *41*(3), 153–166. doi:10.1016/S0303-2647(96)01670-X PMID:9113350

Shields, B. L., Main, J. A., Peterson, S. W., & Strauss, A. M. (1997, September). An Anthropomorphic Hand Exoskeleton to Prevent Astronaut Hand Fatigue During Extravehicular Activities. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans*, *27*(5), 668–673. doi:10.1109/3468.618265 PMID:11541130

Singh, R. M., & Chatterji, S. (2012). Trends and Challenges in EMG Based Control Scheme of Exoskeleton Robots – A Review. *International Journal of Scientific & Engineering Research*, *3*(8).

Soft Exosuit. Lightweight suit to increase the wearer's strength and endurance. (n.d.). Wyss Institute. Retrieved from http://wyss.harvard.edu/viewpage/456

Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., & Paige, R. (2012). Large-scale complex IT systems. *Communications of the ACM*, *55*(7), 71–77. doi:10.1145/2209249.2209268

Song, Q., & Han, J.-D. (2008). An adaptive UKF algorithm for the state and parameter estimations of a mobile robot. *Acta Automatica Sinica*, *34*(1), 72–79. doi:10.3724/SP.J.1004.2008.00072

Song, T., Xi, F. F., Guo, S., & Lin, Y. (2016). Optimization of a Mobile Platform for a Wheeled Manipulator. ASME. *Journal of Mechanisms and Robotics*, *8*(6), 061007–061007, 14. doi:10.1115/1.4033855

Song, T., Xi, F. F., Guo, S., Ming, Z. F., & Lin, Y. (2015). A Comparison Study of Algorithms for Surface Normal Determination Based on Point Cloud Data. *Precision Engineering*, *39*, 47–55. doi:10.1016/j.precisioneng.2014.07.005

Song, T., Xi, F. F., Guo, S., Tu, X. W., & Li, X. H. (2017). Slip Analysis for a Wheeled Mobile Manipulator. ASME. J. Dyn. Sys. *Measurement and Control*, *140*(2), 021005–021005, 12. doi:10.1115/1.4037287

Song, Z., & Guo, S. (2011). Design Process of Exoskeleton Rehabilitation Device and Implementation of Bilateral Upper Limb Motor Movement. *Journal of Medical and Biological Engineering*, *32*(5), 323–330. doi:10.5405/jmbe.987

Song, Z., Lipinski, D., & Mohseni, K. (2017). Multi-vehicle cooperation and nearly fuel-optimal flock guidance in strong background flows. *Ocean Engineering*, *141*, 388–404. doi:10.1016/j.oceaneng.2017.06.024

SPARC The Partnership for Robotics in Europe. (2016). *Robotics 2020. Multi-Annual Roadmap.* Release B 02/12/2016. Author.

Spector, L., Klein, J., Perry, C., & Feinstein, M. (2005). Emergence of collective behavior in evolving populations of flying agents. *Genetic Programming and Evolvable Machines*, *6*(1), 111–125. doi:10.100710710-005-7620-3

Spicher, A., Michel, O., & Giavitto, J. (2005) Algorithmic self-assembly by accretion and by carving in MGS. In *Proceedings of the 7th International Conference on Artificial Evolution (EA '05)* (pp. 189–200). Berlin: Springer-Verlag.

Stergiopoulos, P., Fuchs, P., & Laurgeau, C. (2003). *Design of a 2-Finger Hand Exoskeleton for VR Grasping Simulation*. EuropHaptics.

Taber, L. A. (2004). *Nonlinear Theory of Elasticity: Applications in Biomechanics*. Singapore: World Scientific. doi:10.1142/5452

Taggart, W., Turkle, S., & Kidd, C. (2005). An interactive robot in a nursing home: Preliminary remarks. Workshop: Towards Social Mechanisms of Android Science, Stresa, Italy.

Takadama, K., Sato, F., Otani, M., Hattori, K., Sato, H., & Yamaguchi, T. (2012). Preference clarification recommender system by searching items beyond category. In *Proceedings of IADIS (International Association for Development of the Information Society) International Conference Interfaces and Human Computer Interaction 2012* (pp. 3-10). Lisbon, Portugal: IADIS Press.

Teixeira, T., Dublon, G., & Savvides, A. (2010). A survey of human-sensing: Methods for detecting presence, count, location, track, and identity. *ACM Computing Surveys*, *5*(1), 59–69.

Teixeira, T., Dublon, G., & Savvides, A. (2010). A Survey of Human-Sensing: Methods for Detecting Presence. *Count, Location, Track, and Identity*, *3*(10), 133.

Teo, J. J. Y., Woo, S. S., & Sarpeshkar, R. (2015). Synthetic biology: A unifying view and review using analog circuits. *IEEE Transactions on Biomedical Circuits and Systems*, *9*(4), 453–474. doi:10.1109/TBCAS.2015.2461446 PMID:26372648

Timotheou, S., & Loukas, G. (2009). Autonomous networked robots for the establishment of wireless communication in uncertain emergency response scenarios. *Proceedings of the 2009 ACM Symposium on Applied Computing (SAC '09)*. 10.1145/1529282.1529542

Topaz, C. M., Bertozzi, A. L., & Lewis, M. A. (2006). A nonlocal continuum model for biological aggregation. *Bulletin of Mathematical Biology*, *68*(7), 1601–1623. doi:10.100711538-006-9088-6 PMID:16858662

Tuna, G., Gungor, V. C., & Gulez, K. (2014). An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters. *Ad Hoc Networks*, *13*, 54–68. doi:10.1016/j.adhoc.2012.06.006

Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *237*(641), 37–72. doi:10.1098/rstb.1952.0012

Tzafestas, C. S. (2003). Whole-Hand Kinesthetic Feedback and Haptic Perception in Dexterous Virtual Manipulation. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans*, *33*(1), 100–113. doi:10.1109/TSMCA.2003.812600

Uemura, M., Kanaoka, K., & Kawamura, K. (2006). Power Assist Systems Based on Resonance of Passive Elements. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4316-4321. 10.1109/IROS.2006.281965

Umetani, Y., & Yamada, Y. (1999). *Skil Mate. In Wearable Exoskeleton Robot* (pp. 984–988). IEEE.

Umetani, Y., & Yoshida, K. (2010). Resolved motion rate control of space manipulators with generalized Jacobian matrix. *Robotics & Automation IEEE Transactions on*, *5*(3), 303–314. doi:10.1109/70.34766

US Army to Build Armored Talos Suit That Merges Man and Machine. (n.d.). Singularity HUB.

Vafa, Z., & Dubowsky, S. (1987). On the dynamics of manipulators in space using the virtual manipulator approach. *IEEE International Conference on Robotics and Automation. Proceedings*, 579-585. 10.1109/ROBOT.1987.1088032

Velasco-Alvarez, F., Ron-Angevin, R., & Lopez-Gordo, M. A. (2013). *BCI (Brain Controller Interface) -based Navigation in Virtual and Real Environments* (Vol. 7903). Advances in Computational Intelligence. Springer Lecture Notes in Computer Science.

Vergassola, M., Villermaux, E., & Shraiman, B. I. (2007). 'Infotaxis' as a strategy for searching without gradients. *Nature*, *445*(7126), 406–409. doi:10.1038/nature05464 PMID:17251974

Vesely, W. E. (1981). *Fault Tree Handbook.* US Nuclear Regulatory Committee Report, NUREG-0492.

Vesely, W., Stamatelatos, M., Dugan, J., Fragola, J., Minarick, J. III, & Railsback, J. (2002). *Fault Tree Handbook with Aerospace Applications*. NASA Office of Safety and Mission Assurance.

Viana, M., Alencar, P., & Lucena, C. (2016). A Modeling Language for Adaptive Normative Agents. In *Multi-Agent Systems and Agreement Technologies* (pp. 40–48). Cham: Springer.

Vicsek, T., Czirok, A., Ben-Jacob, E., Cohen, I., & Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, *6*(75), 1226–1229. doi:10.1103/PhysRevLett.75.1226 PMID:10060237

Visinsky, M. L., Walker, I. D., & Cavallaro, J. R. (1993). Robotic fault tolerance: algorithms and architectures. *Robotics and Remote Systems in Hazardous Environments*, 53-73.

Vukobratovic, M., Ciric, V., & Hristic, D. (1972). Contribution to the Study of Active Exoskeletons. *Proceedings of the 5th International Federation of Automatic Control Congress.*

Vysin, M., & Knoflicek, R. (2003). The Hybrid Mobile Robot. *IEEE International Conference on Industrial Technology (ICIT)*, *1*(12), 262–264. doi: 10.1109/ICIT.2003.1290291

Walker, M., & Papadopoulos, Y. (2009). Qualitative temporal analysis: Towards a full implementation of the Fault Tree Handbook. *Control Engineering Practice*, *17*(10), 1115–1125. doi:10.1016/j.conengprac.2008.10.003

Wang, K. M., & Hui, L. (2017). Effectiveness evaluation of Internet of Things-aided firefighting by simulation. *The Journal of Supercomputing*. doi:10.100711227-017-2098-3

Weyns, D. (2017). Software engineering of Self-adaptive systems: an organised tour and future challenges. In Handbook of Software Engineering. Springer.

Wilensky, U. (2005). *NetLogo Flocking 3D Alternate model.* Retrieved from http://ccl.northwestern.edu/netlogo/models/

Wilensky, U. (1999). *NetLogo.* Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University; http://ccl.northwestern.edu/netlogo/

Willis, M. A. (2002). *Biologically-Inspired Search Algorithms for Locating Unseen Odor Sources.* Belangar. doi:10.21236/ADA402125

Winder, S. B., & Esposito, J. M. (2008, March). Modeling and control of an upper-body exoskeleton. In *System Theory, 2008. SSST 2008. 40th Southeastern Symposium on* (pp. 263-268). IEEE. 10.1109/SSST.2008.4480234

Winfield, A. F. T., Sav, J., Fernández-Gago, M.-C., Dixon, C., & Fisher, M. (2005). On formal specification of emergent behaviours in swarm robotic systems. *International Journal of Advanced Robotic Systems*, *2*(4), 363–370. doi:10.5772/5769

Xi, F. F. (2009). *Computational Dynamics (Graduate Course Lecture Notes)*. Toronto, ON, Canada: Ryerson University.

Yakymets, N., Dhouib, S., Jaber, H., & Lanusse, A. (2013). Model-driven safety assessment of robotic systems. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on* (pp. 1137-1142). IEEE.

Yamaguchi, T., Nishimura, T., & Takadama, K. (2012). *Awareness based recommendation - Toward the Cooperative Learning in Human Agent Interaction*. Paper presented at The International Conference on Humanized Systems 2012 (OS02_1003). Daejeon, South Korea.

Yamaguchi, T., Nishimura, T., & Sato, K. (2011). How to recommend preferable solutions of a user in interactive reinforcement learning? In A. Mellouk (Ed.), *Advances in Reinforcement Learning* (pp. 137–156). Rijeka, Croatia: InTech Open Access Publisher. doi:10.5772/13757

Yamaguchi, T., Nishimura, T., & Takadama, K. (2009). Awareness based filtering - Toward the Cooperative Learning in Human Agent Interaction. In *Proceedings of the ICROS-SICE International Joint Conference (ICCAS-SICE 2009)* (pp. 1164-1167). Tokyo, Japan: The Society of Instrument and Control Engineers.

Yamaguchi, T., Takemori, K., & Takadama, K. (2013). Modeling a human's learning processes toward continuous learning support system. In M. K. Habib & J. P. Davim (Eds.), *Mechatronics Engineering* (pp. 69–94). Hoboken, NJ: Wiley-ISTE.

Yamaguchi, T., Takemori, K., Tamai, Y., & Takadama, K. (2015). Analyzing human's continuous learning processes with the reflection sub task. *Journal of Communication and Computer*, *12*(1), 20–27.

Yamins, D. (2005). Towards a theory of "local to global" in distributed multi-agent systems. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMS 2005)* (pp. 183–190). Utrecht, The Netherlands: Academic Press. 10.1145/1082473.1082501

Yamins, D. (2007). *A Theory of Local-to-Global Algorithms for One-Dimensional Spatial Multi-Agent Systems* (PhD thesis). Cambridge, MA: Harvard University.

Yamins, D., & Nagpal, R. (2008). Automated global-to-local programming in 1-D spatial multi-agent systems. In L. Padgham, D. C. Parkes, J. P. Müller, S. Parsons (Eds.), *Proceedings of 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)* (pp. 615–622). Estoril, Portugal: Academic Press.

Yan, Z., Jouandeau, N., & Ali, A. (2013). A Survey and Analysis of Multi-Robot Coordination. *International Journal of Advanced Robotic Systems*, *1*. doi:10.5772/57313

Yoshida, K. (2009). Achievements in Space Robotics Expanding the Horizons of Service and Exploration. *IEEE Robot Autom Mag, 16*, 20-28.

Yuan, J., Tang, S., Wang, C., De, D., Li, X., Song, W., & Chen, G. (2011). A real-time rescue system: Towards practical implementation of robotic sensor network. *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON).* doi:10.1109ahcn.2011.5984930

Yuan, P., Wang, T., Ma, F., & Gong, M. (2014). Key Technologies and Prospects of Individual Combat Exoskeleton. Knowledge Engineering and Management, 305-316. doi:10.1007/978-3-642-37832-4_28

Yu, Z. W., Liu, X. F., & Cai, G. P. (2016). Dynamics modeling and control of a 6-DOF space robot with flexible panels for capturing a free floating target. *Acta Astronautica*, *128*, 560–572. doi:10.1016/j.actaastro.2016.08.012

Zarzhitsky, D., Spears, D., Thayer, D., & Spears, W. (2004, April). Agent-based chemical plume tracing using fluid dynamics. In *International Workshop on Formal Approaches to Agent-Based Systems* (pp. 146-160). Springer Berlin Heidelberg. 10.1007/978-3-540-30960-4_10

Zeilig, G., Weingarden, H., Zwecker, M., Dudkiewicz, I., Bloch, A., & Esquenazi, A. (2012). Safety and Tolerance of the ReWalk™ Exoskeleton Suit for Ambulation by People with Complete Spinal Cord Injury: A Pilot Study. *The Journal of Spinal Cord Medicine*, *35*(2), 101–196. doi:10.1179/2045772312Y.0000000003 PMID:22333043

Zhang, Y., Zhang, J., Hao, G., & Zhang, W. (2015, August). Localizing odor source with multi-robot based on hybrid particle swarm optimization. In *Natural Computation (ICNC), 2015 11th International Conference on* (pp. 902-906). IEEE. 10.1109/ICNC.2015.7378110

Zhang, B., Liang, B., Wang, Z. W., Mi, Y. L., Zhang, Y. M., & Chen, Z. (2017). Coordinated stabilization for space robot after capturing a noncooperative target with large inertia. *Acta Astronautica*, *134*, 75–84. doi:10.1016/j.actaastro.2017.01.041

Zhang, Y. L., Ma, X. P., & Miao, Y. Z. (2013). A Virtual Physics-Based Approach to Chemical Source Localization Using Mobile Robots. *Applied Mechanics and Materials*, *263*, 674–679.

Zhu, H., & Lightfoot, D. (2003, August). Caste: A step beyond object orientation. In *Joint Modular Languages Conference* (pp. 59-62). Springer. 10.1007/978-3-540-45213-3_8

Zhu, H., & Zhou, M. (2006). Role-based collaboration and its kernel mechanisms. *IEEE Transactions on Systems, Man and Cybernetics. Part C, Applications and Reviews*, *36*(4), 578–589. doi:10.1109/TSMCC.2006.875726

Zielinska, T. (2006). Control and navigation aspects of a group of walking robots. *Robotica*, *24*(1), 23–29. doi:10.1017/S0263574705001840

Zielinski, C. (2010). Robotics - Quo vadis? *Journal of Measurements and Control, 5*, 9-19. (in Polish)

Zielinski, C., Stefańczyk, M., Kornuta, T., Figat, M., Dudek, W., Szynkiewicz, W., ... Iturburu, M. (2017). Variable structure robot control systems: The RAPP approach. *Robotics and Autonomous Systems*, *94*, 226–244. doi:10.1016/j.robot.2017.05.002

Zoss, A. B., Kazerooni, H., & Chu, A. (2006, April). Biomechanical Design of the Berkeley Lower Extremity Exoskeleton (BLEEX). *IEEE/ASME Transactions on Mechatronics*, *11*(2), 128–138. doi:10.1109/TMECH.2006.871087

Zoss, A., & Kazerooni, H. (2006). Design of an Electrically Actuated Lower Extremity Exoskeleton. *Advanced Robotics*, *20*(9), 967–988. doi:10.1163/156855306778394030

Zou, Y., & Luo, D. (2008). A modified ant colony algorithm used for multi-robot odor source localization. *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, 502-509.

About the Contributors

Sarah Allali is a PhD student with the mobility team of the LSI laboratory, at the University of Science and Technology Houari Boumediene (USTHB), Algiers. Her main research interests are on wireless sensor networks and robotic systems for search and rescue operations. She obtained a MSc degree in Networking, Telecommunications and System from University Claude Bernard Lyon 1, France, in 2013. She also holds a MSc degree in Computer Networks and Distributed Systems from USTHB.

Mahfoud Benchaïba received the Computer Science Engineer Degree from the University of Batna, Algeria in 1988. He received the Master Degree in computer science at the University of Sciences and Technologies Houari Boumediène (USTHB) at Algiers in 1992. He received the PhD Degree in Computer Science from the USTHB in November 2, 2009. He is actually full professor in the Computer Science Department at USTHB. He is a Research Director and is a member of Mobility Team in LSI Laboratory at USTHB. His research interests include Distributed Systems, Mobile Ad hoc Networks, Wireless sensor (and robot) networks, Wireless body area networks, Distributed Algorithms and services, P2P systems.

Gangqi Dong received his B.Sc. and M.Sc. degrees in guidance, navigation, and control from Northwestern Polytechnical University, Xi'an, Shaanxi, China, in 2005 and 2008, respectively. He received his PhD degree in earth and space science from York University in Canada in 2016. He was with Xi'an Institute of Microelectronic Technology as a Computer Hardware Engineer from 2008 to 2010. Then he joined Huawei Technologies Co., Ltd. as a Hardware engineer of communication from 2010 to 2011. From 2011 to 2016, he was a PhD student at York university in Canada. He is currently an assistant Professor with Northwestern Polytechnical University, Xi'an, Shaanxi, China. His research interests include development of mechatronic systems and visual servo robotic systems.

Menggao Dong is an Assistant Professor of the College of Intelligence Science and Technology, National University of Defense Technology, China. His research interests include Self-Adaptive Systems, Multi-Agent Systems, Software Engineering, Reinforcement Learning, and Information Extraction Techniques. He has published 19 research papers.

W. Wilfred Godfrey is an assistant professor at ABV-Indian Institute of Information Technology & Management Gwalior, MP, India. His research interests are AI-Robotics, Embedded Systems, Machine Learning and Multi Agent Systems.

Shuai Guo received the Ph.D. degree in mechanical engineering from Shanghai University, Shanghai, China. He is an Associate Professor of mechanical engineering at Shanghai University. His areas of interest and expertise include robotics, autonomous mobile robots, and mechanization system modeling. He has authored more than 45 journal and conference papers in the areas of controlled mechanical systems, rehabilitation, and robotics.

Upma Jain received the M.tech. degree in computer science from ABV-IIITM, Gwalior in 2012. Currently she is Ph.D. candidate in ABV-IIITM Gwalior. Her research interest includes Robotics, Artificial Intelligence and Soft Computing and Applications.

Bruce MacLennan received his BS (mathematics, honors) from Florida State University in 1968 and his MS and PhD (computer science) from Purdue University in 1972 and 1975. After receiving his PhD, he worked for Intel Corporation (1975–9), after which he joined the Computer Science faculty of the Naval Postgraduate School (Monterey, CA) as Assistant Professor (1979–83), Associate Professor (1983–7), and Acting Chair (1984–5). Since 1987 has been an Associate Professor in the Department of Electrical Engineering & Computer Science of the University of Tennessee, Knoxville. Since the mid-1980s, his research has focused on new approaches to self-organization, robotics, and artificial intelligence based on neuroscience. Prof. MacLennan has more than 90 refereed journal articles and book chapters and has published three books. He has made more than 70 invited or refereed presentations.

Nidhal Mahmud works in the fields of model-based analysis and design of system architectures. He holds a DEA in Conception of the Advanced Informatics Systems from the University of Évry Val d'Essonne (France) and a PhD degree in

Computer Science from the University of Hull (UK). His research is focused on safety and reliability of complex real-time embedded systems and he has been involved in the research and development of EAST-ADL, an engineering approach for automotive electronic systems. Previous EU project involvements include ATESST2 and MAENAD.

Xinjun Mao is a Full Professor and Director of Intelligent Software Laboratory at National University of Defense Technology, China. He received his PhD degree of computer science from National University of Defense Technology in China in 1998. He was a visiting professor of University of Toronto, Oxford Brookes University during 2003-2014. He is a Distinguished Member of CCF and member of IEEE and ACM. His research interests include multi-agent systems, self-adaptive and self-organizing software, crowdsourcing-based software engineering, and autonomous robot software. He has published more than 150 research papers in academic conferences and journals, five books, and three book chapters in Springer and Tsinghua Press. He also has experiences in industry software and cooperation with famous IT companies like Huawei in China. He has served as Program Chair or Member of more than 40 academic conferences such as AAAI, AAMAS, IJCAI, APSEC, PRICAI, SOSE, IEEE SMC, ICA3PP, and member of editorial board for several international academic journals.

Shota Nagahama is a graduate student of the Faculty of Advanced Engineering, National Institute of Technology, Nara College, Japan. His research interests include reinforcement learning, parallel algorithm, music information retrieval, learning support system and recommendation system.

Takuma Nishimura received his M.E. degree from Kyoto University, Japan, in 2012. He joined NTT WEST from 2012. His research interests include museology, influential medium, recommender system, multiagent system, autonomous system, human-agent interaction, reinforcement learning, and emergent computation.

Thomas M. Schnieders received his MS in Human Computer Interaction and Industrial Engineering from Iowa State University in 2016, where he also received his BS in Mechanical Engineering. He is the co-founder and co-director of The ATHENA Lab at Iowa State University where he is a PhD candidate in Industrial Engineering. His research revolves around human performance enhancement, human augmentation, and the design and manufacture of exoskeletons as it pertains to augmentation and training.

Tao Song received the Ph.D. degree in mechanical engineering from Shanghai University, PRC. He is a post-doctor of control engineering at Shanghai University. His areas of interest and expertise include robotics, mobile manipulator and mechanization system modeling.

Richard T. Stone received his PhD in Industrial Engineering from the State University of New York at Buffalo in 2008. Dr. Stone is currently an Associate Professor at Iowa State University in the department of Industrial and Manufacturing Systems Engineering. The core of Dr. Stone's research is in human augmentation and human performance enhancement in both physical and mental domains. He employs multiple approaches towards this goal, including cognitive and physiological engineering, classical and experimental ergonomics, augmented reality, and the incorporation and application of new technologies. Dr. Stone's past research has involved the development of design methodologies for AR and multisensory devices, telerobotic control system development, the application of biomechanics for improved sports performance and the development of visualization tools for improving battle space awareness. In addition to being a Professor Dr. Stone is a Sheriff's Deputy (reserve unit) and Recovery Diver for the Story County Sherriff's Office.

Keiki Takadama got his Ph.D. Degree from the University of Tokyo, Japan, in 1998. He joined Advanced Telecommunications Research Institute (ATR) International from 1998 to 2002 as a visiting researcher and worked at Tokyo Institute of Technology from 2002 to 2006 as a lecturer. He moved to The University of Electro-Communications as an associate professor in 2006 and is currently a professor of the Department of Informatics from 2011. His research interests include multiagent system, human-agent interaction, reinforcement learning, evolutionary computation, sleep and healthcare system, and autonomous system in space. He is a member of IEEE, ACM, and a member of major AI- and informatics-related academic societies in Japan.

Ritu Tiwari is an Assistant Professor in the ICT Department of ABV-Indian Institute of Information Technology and Management, Gwalior, India. Her field of research includes bio-metrics, artificial neural networks, signal processing, robotics and soft computing. She has authored or coauthored more than 50 technical articles and 4 books.

Fengfeng (Jeff) Xi received thePh.D. degree in robotics and automation in 1993 from the University of Toronto and was awarded a NSERC Postdoctoral Fellowship with which he worked as a Postdoctor at McGill University and the University of Waterloo.His research interests are in the area of manufacturing and automation. His current research in manufacturing focuses on micro manufacturing methods including abrasive machining, laser processing, and precision forming, with applications to aerospace and automotive industries. The ongoing work includes process modeling and simulation, and new process development. In particular, a prototype computer aided manufacturing (CAM) software package is being developed that can simulate the surface roughness of the workpiece produced by abrasive machining including grinding, polishing, super-finishing and honing. His current research in automation focuses on the development of robotic systems to automate manual manufacturing processes such as polishing and riveting.

Tomohiro Yamaguchi received his M.E. degree from Osaka University, Japan, in 1987. He joined Mitsubishi Electric Corporation in 1987 and moved to Matsushita Electric Industrial in 1988. He worked at Osaka University from 1991 to 1998 as a research associate and got Doctor of Engineering Degree from Osaka University in 1996. He moved to Nara National College of Technology as associate professor in 1998 and is currently a professor from 2007. His research interests include interactive recommender system, music information retrieval, multiagent reinforcement learning, autonomous learning agent, human-agent interaction, learning support system, human learning process and mastery process. He is a member of The Japanese Society for Artificial Intelligence and The Society of Instrument and Control Engineers, Japan.

Haibin Zhu is a Full Professor and the coordinator of the Computer Science Program, Founding Director of Collaborative Systems Laboratory, Nipissing University, Canada. He has published 160+ research papers, five books and four book chapters. He is a senior member of IEEE and is serving and served as co-chair of the technical committee of Distributed Intelligent Systems of IEEE SMC Society, associate editor of IEEE SMC Magazine, guest (co-) editor for 3 special issues of prestigious journals, and organization chairs for many IEEE conferences. He was a Program Committee (PC) Chair for 17th IEEE Int'l Conf. on Computer Supported Cooperative Work in Design, Whistler, BC, Canada, June 27- 29, 2013. He also served as PC members for 60+ academic conferences. He is a founding researcher of Role-Based Collaboration and Adaptive Collaboration. He is the receipt of the chancellor's award for excellence in research (2011) and two research achievement awards from Nipissing University (2006-2007, 2012-2013), the IBM Eclipse Innovation Grant Awards(2004, 2005), the Best Paper Award from the 11th ISPE Int'l Conf. on Concurrent Engineering (ISPE/CE2004), the Educator's Fellowship of OOPSLA'03, a 2nd class National Award for Education Achievement(1997), and three 1st Class Ministerial Research Achievement Awards from China (1997, 1994, and 1991). His research interests include Collaboration Theory, Technologies, Systems, and Applications (Role-Based Collaboration and Adaptive Collaboration), Human-Machine Systems, CSCW (Computer-Supported Cooperative Work), Multi-Agent Systems, Software Engineering, and Distributed Intelligent Systems.

Zheng H. Zhu is a professor and York Research Chair in Space Technology - Tier 1 at the York University in Toronto, Canada. He received his B.Eng., M.Eng. and Ph.D. degrees in mechanics from Shanghai Jiao Tong University located in Shanghai, China. He also received the M.A.Sc. in robot control from University of Waterloo and Ph.D. in mechanical engineering from University of Toronto all located in Ontario, Canada. After two years (1993-1995) as a research associate at the Department of Mechanical and Industrial Engineering of University of Toronto, he spent 11 years in aerospace at Indal Technologies - Curtiss-Wright Defense Solutions in Mississauga, Ontario. He joined the Department of Earth and Space Science and Engineering at the York University in 2006 till 2016. Currently, his is a Professor at the Department of Mechanical Engineering at the York University. Professor Zhu's research interests include dynamics and control of tethered space system, space service robot and nanotechnology-based multifunctional materials for spacecraft. He has published over 240 articles and conference papers. Professor Zhu is the fellow of Engineering Institute of Canada, Fellow of CSME and ASME, Associate fellow of AIAA, senior member of IEEE, and licensed Professional Engineer in Ontario, Canada. He is the Editor-in-chief of the International Journal of Space Science and Engineering, associate editor of IEEE Access, and a member of editorial boards of multiple international journals.

Teresa Zielinska received her MSc in Engineering and PhD in Robotics from the Warsaw University - Poland. She obtained D.Sc. (habilitation) from Polsh Academy of Sciences in 1995, in 2005 she awarded the research title of professor in the field of technical sciences. She is full professor at the Warswa University of Technology. Her research interest covers robot control systems, novel concepts of biologically inspired robots, motion synthesis of walking machines and humanoids, and SLAM methods for mobile robotics She writes and presents widely on issues of service robotics.

Index