Aalborg Universitet

AALBORG UNIVERSITY

Interactions Between Humans and Robots

Vlachos , Evgenios ; Schärfe, Henrik

Publication date:
2013

Document Version
Early version, also known as pre-print

Link to publication from Aalborg University

Citation for published version (APA):
Vlachos , E., & Schärfe, H. (2013). *Interactions Between Humans and Robots*. Abstract from 2012 AAU Workshop on Human-Centered Robotics, Aalborg, Denmark.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: May 16, 2025

AALBORG UNIVERSITY
DENMARK

1st AAU Workshop on Human-Centered Robotics

November 15, 2012, Aalborg University, Denmark




Proceedings

Edited by Shaoping Bai, Jesper A. Larsen, Ole Madsen, Matthias Rehm

AALBORG UNIVERSITY PRESS

1st AAU Workshop on Human-Centered Robotics
Edited by Shaoping Bai, Jesper A. Larsen, Ole Madsen and Matthias Rehm

© The authors and Aalborg University Press 2013

Open Access publication

Cover pictures from left:

- Light weight robotic arm, www.en.m-tech.aau.dk/Research+Groups/Robotics+and+Automation
- Little helper, www.en.m-tech.aau.dk/Research+Groups/Robotics+and+Automation
- Prosthetic hand illustration, photo by Dalila Lepirica and Shellie Boudreau
- Henrik & Henrik, http://www.hum.aau.dk/~scharfe/

ISBN: 978-87-7112-090-5

Published by:
Aalborg University Press
Skjernvej 4A, 2nd floor
DK – 9220 Aalborg
Denmark
Phone: (+45) 99 40 71 40
aauf@forlag.aau.dk
http://forlag.aau.dk/forside.aspx

1st AAU Workshop on Human-Centered Robotics

The 2012 AAU Workshop on Human-Centered Robotics took place on 15 Nov. 2012, at Aalborg University, Aalborg. The workshop provides a platform for robotics researchers, including professors, PhD and Master students to exchange their ideas and latest results. The objective is to foster closer interaction among researchers from multiple relevant disciplines in the human-centered robotics, and consequently, to promote collaborations across departments of all faculties towards making our center a center of excellence in robotics. The workshop becomes a great success, with 13 presentations, attracting more than 45 participants from AAU, SDU, DTI and industrial companies as well.

The proceedings contain 7 full papers selected out from the full papers submitted afterwards on the basis of workshop abstracts. The papers represent major research development of robotics at AAU, including medical robots, social, serving robots and HRI, and innovative robot design, control and vision technologies as well.

The full papers were reviewed by a board of external reviewers, which include:

- Dr. Stephane Caro, Institut de Recherche en Communications et Cybernetique de Nantes (IRCCyN), France
- Professor Heping Chen, Texas State University-San Marcos, Texas, US
- Professor Tomoko Koda, Dept. of Information Science and Technology, Osaka Institute of Technology, Osaka, Japan
- Professor Markus Vincze, Institut für Automatiserungs- und Regelungstechnik, TU Wien, Austria
- Professor Antonios Gasteratos, Dept. of Production and Management Engineering, Democritus University of Thrace, Xanthi, Greece
- Professor Yukiko Nakano, Dept. of Computer and Information Science, Seikei Univeristy, Tokyo, Japan

Their time and effort to ensure the quality of the proceedings are greatly appreciated.

We wish you an interesting reading!

The Workshop Organizing Committee

Ole Madsen, Matthias Rehm, Jesper A. Larsen, Shaoping Bai

Programme

9:00-9:10 Welcome, Shaoping Bai

9:10-10:30 Keynotes, Human-Centered Robotics; Henrik Scharfe, Ole Madsen, and Matthias Rehm

10:30-10:45 Coffee Break

10:45-12:30 Session I--Advanced Robots

- **1. Safe Surgical Robots; C. Sloth, R. Wisniewski, J. A. Larsen, J. Leth, and J. Poulsen**
- **2. Implementing Force-feedback in a telesurgery environment using parameter estimation,**
  T. Hansen, et al. , Electronics Systems, AAU
- **3. Design of a Passive Exoskeleton; L. Zhou, S. Bai, J. Rasmussen, M-Tech, AAU**
- **4. On Autonomous Monitoring Platforms; R. Christensen, N. Østergaard and J. A. Larsen**
- **5. Human Assisted Instructing of AIMM Robot; C. Carøe, C. Schou, M. Hvilshøj, S. Bøgh and O.Madsen, M-Tech, AAU**

12:30-13:30 Lunch Break

13:30-15:00 Session II—Human-robot Interaction

- **6. Giraff - a mobile robot for the home; S. Von Rump, Giraff Technologies AB**
- **7. Interactions between Humans and Robots; E. Vlachos and H. Scharfe, AAU**
- **8. PEERs: Persuasive Education & Entertainment Robotics; L. B. Bertel, Technology Institute**
- **9. On Gender Differences in the Perception of a Geminoid; J. Abildgaard and H. Scharfe , AAU**

15:00-15:20 Coffee Break and Posters

15:20-16:50 Session III—Visions and motion planning

- **10. A Human-Robot Interaction Application for the Localization of Objects Based on the Evaluation of Unconstrained Pointing Gesture with the Use of a Depth Camera; V. Krüger and H. Karafiat**
- **11. Real-Time Image Segmentation Using a Fixation-Based Approach ; V. Krüger and B. Grossmann**
- **12. Design and Locomotion Mode Analysis of a Radial Symmetrical Six-wheel-legged Robot, K. Xu, X. Ding, and S. Bai, Beihang University, China & M-Tech, AAU**
- **13. Human Assisted Computer Vision on Industrial Mobile Robots; R. S. Andersen, O. Madsen, T. B. Moeslund, M-Tech and Media Technology, AAU**

16:50-17:00 Conclusion

Papers

Persuasive Educational and Entertainment Robotics (PEERs)

Lykke Brogaard Bertel
Department of Communication and Psychology
AAU Aalborg University
Aalborg, Denmark
lbb@hum.aau.dk

***Abstract*— This paper explores and develops the concept of Persuasive Educational and Entertainment Robotics (PEERs) as a theoretical framework for designing robots to facilitate motivation for play and learning in schools.**

Keywords— Persuasive Design, Human-Robot Interaction, Didactic design

I. INTRODUCTION

According to EUROP's Strategic Research Agenda for robotics in Europe, education and entertainment will be among the primary areas of application for the advanced robotics of tomorrow [1]. Robotic trainers, teachers and playmates will interact with us physically and socially and often act as a medium – a platform for communication, experimentation and collaboration. Research within Human-Robot Learning or *r-learning* show that minimally expressive and thus easily readable robots can be used to teach children with an autism diagnosis how feelings are expressed, experienced and interpreted [2-3]. In addition, an increasing number of studies within Human-Robot Interaction (HRI) explore the use of robots as educational technologies in teaching specific curricula such as science [4] and language learning [5]. These results suggest that robots, in addition to their intrinsic entertainment value, do have a certain ability to facilitate motivation for learning. However, few studies have been conducted on how the design of these robots (e.g. physical appearance, interaction patterns and communication modalities etc.) mediate motivation and how this can be used strategically to facilitate learning. Thus, a demand for heuristics for designing robots to motivate play and learning has emerged. This paper argues that the combination of research within Human-Robot Interaction, Persuasive Design and didactics can provide a theoretical framework for this specific purpose.

In the following section, key principles within the theoretical framework of HRI, Persuasive Design and didactics are presented. In section III the concept of Persuasive Educational and Entertainment Robotics (PEERs) is introduced and related to the previously presented principles. In section IV, the applicability of PEERs to the Danish educational system is discussed and directions for future research within PEERs are proposed.

II. THEORETICAL FRAMEWORK

In this section the theoretical frameworks of Persuasive Design, Human-Robot Interaction and didactics are briefly introduced. Their contribution to the development of PEERs is specified and shortcomings identified.

1. Persuasive Design

Persuasive Technology as a concept defining technologies that aim to motivate attitude or behavior change was first introduced in 1998 [6] and established as its own field of research in 2003 [7]. Persuasive technology is defined as *"any interactive computing system designed to change people's attitudes or behaviors or both without using coercion or deception"*, and the research field of Persuasive Design (PD) thus concerns specifically with the design and development of these technologies, e.g. to promote a healthier or more sustainable lifestyle and thus increase the quality of life for the user, the society and the environment. The theoretical framework of Persuasive Design revolves around the Functional Triad, which describes three different roles that the technology can take on and utilize when attempting to motivate behavior change [7]. A persuasive technology can thus act as a:

1. Tool; *simplifying or guiding tasks, tailoring the interaction to the user, providing the possibility of self-monitoring and surveillance, suggesting and rewarding behavior*
2. Medium; *providing compelling experiences and the opportunity to explore complex cause/effect relationships through the simulation of environments or objects*
3. Social Actor; *interacting with the user socially, providing feedback and social support, gaining trust through similarity or authority and eliciting reciprocity* (i.e. similar to what has elsewhere been termed *relational agents* [8])

Although emphasizing robots as natural social actors, few studies within PD investigate the persuasiveness of feedback from a robot as opposed to other types of social support [9]. The concept of PEERs will thus need to include reflections regarding persuasiveness compared to both the teacher and other technologies within the educational context.

2. Human-Robot Interaction

In 2003, Socially Interactive Robotics was introduced to define robots whose sole purpose is to engage in social interactions and the following taxonomy of the characteristics and the key components of HRI was developed [10]:

TABLE I. SOCIALLY INTERACTIVE ROBOTS (SIR)

Socially Interactive Robotics		
Attribute	***Description***	***Example***
Morphology	Establishes social expectations of the interaction and provides information about the intended use and function of the robot	Anthropomorphic Zoomorphic Caricatured Functional
Emotions	Facilitate credibility in HRI and serve as feedback to the user about the robots internal state.	Anger, fear, sadness, joy, surprise, neutral and *combinations*
Dialogue	Exchange and interpretation of symbols and information about the content and context of the interaction	Synthetic language Natural language Non-verbal cues
Personality	A set of qualities which are particularly significant for a specific robot.	Tool (reliable) Pet (lovable) Character Supernatural Human-like
Perception	In addition to features such as localization, navigation and obstacle avoidance, a social robot must possess number of perceptual abilities to engage in social interaction with humans	Face/gaze tracking Speech/Gesture recognition Tone of voice
User Modeling	The ability to create different user models so as to adapt to and shape the interaction in relation to a specific user characteristics	Qualifications Experience Cognitive abilities
Situated Learning	The ability to transfer information, skills and tasks between robots and humans	Imitation Learning
Intentionality	For people to be able to assess and predict behavior in HRI, it is necessary that the robot expresses intentionality	Targeted behavior Theory of Mind Joint attention

Although providing an overview of key components in HRI, this taxonomy does not evaluate the different characteristics of social robots in relation to the context of the interaction, and does not provide a framework for prioritizing components when designing robots for a specific purpose. Thus, the concept of PEERs must include reflections regarding the strategic choice of components in HRI when designing specifically for play and learning.

3. Didactics

Learning as a concept can be considered both a *product*, i.e. the knowledge, skills, attitudes or values that the learner acquires through experience or education, and a *process* through which such product is obtained. In this context, didactics relate to the initiation and maintenance of this particular process. [11]. It is, however, possible to study didactics from many different paradigms and perspectives with as many different definitions of learning, which in turn reflects different concepts of knowledge and motivation. Thus, this section provides only a brief introduction to the three main schools; behaviorism, cognitivism, and constructivism and argues the latter as the basis for the development of PEERs.

Behaviorism developed by Thorndyke (1887-1949) and Skinner (1904-1990), deals as a psychological discipline only with what is observable, i.e. behavior. Behaviorists believe human behavior to be predictable and mediated solely by external consequences and not by internal motives [12]. Thus, a behavioral approach to teaching will be based on pre-determined goals and a systematic didactic design divided into sub processes, which may be the subject of reward. Today behaviorism is often criticized as dehumanizing and the behavioristic perspective on learning often rejected in didactic design [13]. Yet many educational institutions bear a relic of this early approach to learning, often through the strategic use of "playtime" as a reward and "learning" as punishment (detention). It is argued, though, that the otherwise relatively mechanical behaviorist learning process can be made dynamic through increased reflection on conditioning as something we ourselves orchestrate, as is the case with self-reward [14].

Cognitivism rejects the idea that a teacher can "load" knowledge onto students through various stimuli, and instead considers learning as a mental process through which a learner constructs new knowledge. Jean Piaget's (1896-1980) stage theory is one of the most important within the field of cognitivism [15]. In this theory existing knowledge is organized as a series of schemes, which are used to interpret new information either by *assimilation* (unconscious adaptation of the outside world to existing understandings and schemes) or *accommodation* (rejection and further development of schemes to make sense of the world). According to Piaget's developmental psychology these activities are mutually dependent and constantly interacting as part of the learning process, but the concepts have since been used to identify different forms of learning. The assimilative learning is thus articulated as the hallmark of traditional teaching, whereas project-based learning has been highlighted as significant accommodative learning [16].

Constructivism also views learning as a construction, but in contrast to cognitivism rejects the idea, that psychological development is a progression through stages and emphasizes social interaction as the basis for learning rather than mental processes. Lev Vygotsky (1896-.1934) is often referred to as the founder of the constructivist approach to learning. He argued that social interaction has a profound effect on cognitive development, and thus that learning is mediated through social interaction between the learner and a so-called *more knowledgeable other* and defines the *zone of proximal*

development as the distance between the current capacity of problem solving and the potential capacity for problem solving under the guidance of or in collaboration with such *more knowledgeable others* [17]. With a constructivist approach to learning, developing daily didactic designs thus requires a great commitment from the more knowledgeable other in addition to a profound knowledge of and ability to adapt to the student's individual zone of proximal development. When designing robots that interact with us physically and socially to motivate play and learning, it is valuable to view learning as a process of social interaction between the learner and a more knowledgeable other, and the persuasive potential of the didactic design as mediated by and affecting the zone of proximal development.

III. PEER: REVIEWED

Based on the above reflections on the strengths and shortcomings in the existing theoretical frameworks of HRI, PD and didactics it is argued, that the combination of the principles within these fields of research provides a new and useful framework for designing robots specifically to motivate play and learning. In addition, the PEERs model (fig. 1) can categorize related technologies; A. *Persuasive Robotics*; B. *Educational Robotics*, and C. *Persuasive Learning Designs*. This creates the opportunity to compare these research fields theoretically and compile their respective design strategies and principles when designing robots specifically with the purpose of motivating play and learning in schools.

1. Persuasive Robotics

This term covers the group of robots designed to motivate behavior change through social support. This is also referred to as Socially Assistive Robotics (SAR) [18], which is most often used to describe assistive robotic technologies within rehabilitation supporting training not by physical manipulation but through social interaction. In addition to the existing taxonomy for social robots, the SAR framework includes the following concepts; *User group; Task; Modality* and *Role* [18]. It can be argued, that these concepts defines the overlap between PD and HRI which distinguishes SAR from other types of social robots, since having a predefined user group and task entails a specific (persuasive) intention within the design. These taxonomic additions contribute to the PD-concept of technologies as social actors, as they describe some of the benefits that robots have in the intervention when compared to humans, such as the use of different modalities (text, audio, image, speech and gesture), or the ability to assume different roles depending on the user and task. For instance, in the field of PD, technologies can take on the role as an *authority* to elicit trust and thus motivation. However, in some cases, realizing a persuasive intention requires an inverse relationship between user and robot, as is the case with the robotic seal Paro, which motivates behavior change by taking on the role as the care-*receiver* when interacting with people suffering from dementia, and thus providing these otherwise solely care-receiving users the unique opportunity to take on the role as the care-*giver*, providing purpose and thus increasing quality of life [19].

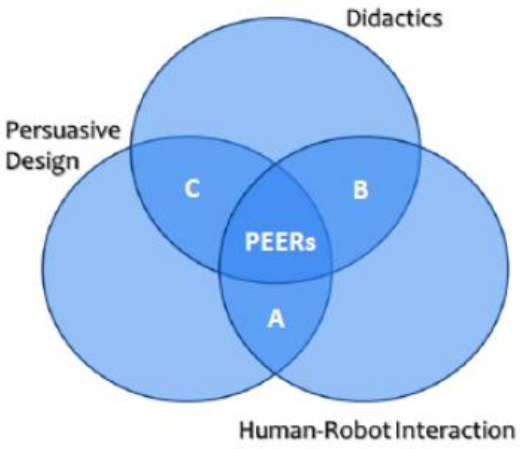


Fig. 1. Persuasive Educational and Entertainment Robotics (PEERs) terms the intersection between Persuasive Design, Human-Robot Interaction and didactics. Related fields are; A. Persuasive Robotics; B. Educational Robotics; and C. Persuasive Learning Designs

2. Educational Robotics

This term represents the relatively small range of robotic devices currently used in schools for educational purposes. Most often these are modular robots or robotic kits such as LEGO Mindstorms [20] used to teach topics within Science, Technology, Engineering, and Math (STEM). These robots are often highly adaptable, rebuildable and programmable and thus facilitate experimentation, reflection and collaboration [21]. Research within this application of robotics focuses primarily on the embodied learning that such educational robots provide and the way in which abstract theoretical concepts can be translated through physical interaction and bodily experience [22].

3. Persuasive Learning Designs

This concept share a common goal with what has been termed *e-learning*, *technology-enhanced learning* [23] or *computer-supported collaborative learning* [24], i.e. the goal of enhancing learning through the use of ICT tools. In this context, Persuasive Learning Designs (PLD) specifically deals with the strategic design of Human-Computer Interaction (HCI) to facilitate *motivation for learning*. Different didactic views and paradigms affect the way in which such design can be approached, and the vast influence that behaviorism has traditionally had on teaching is also reflected in many didactic ICT tools, often based on pre-determined goals and a systematic (and somewhat rigorous) training process concluded with a reward. However, it has been argued, that constructive alignment can also be obtained through a constructivist rethinking of persuasive and didactic principles such as praise and rewards [25, 26]. Concepts within PLD such as persuasive learning objects and technologies are being explored in depth in [27] and theories and methodologies of PLD have been developed in [28].

IV. Discussion and Directions for Future Work

The increasing use of persuasive learning technologies in schools provides the opportunity of utilizing strategies associated with HCI such as multimodality and adaptivity to strengthen teaching within each individual's zone of proximal development. However, not all challenges can be met with screen-based persuasive learning designs. These devices are often personal and highly individualized and thus may not facilitate and support the development of important skills such as communication and collaboration. In this relation, the potential of PEERs is the possibility that robotic technologies in addition to the persuasive (technological) strategies are also able to exploit strategies otherwise associated with and limited to persuasion in the interaction between humans. Obviously, social robots can utilize the persuasive strategies related to being a social actor, but with the expansion of the HRI taxonomy to include distribution of roles between robots and humans, robots can be persuasive in the way in which they break down otherwise rigid, social constructions and structures and create opportunities for new relations and new knowledgeable others.

To explore the applicability of PEERs in the Danish educational system, its design theories and methodologies of implementation are currently being further developed in larger-scale, long-term and cross-contextual case studies in real-world educational settings. These studies focus on the persuasive potential of PEERs to facilitate interdisciplinary exploration, collaboration and reflection and particularly the possibility of PEERs as means of both differentiating education and gathering students in communities of play and learning across individual needs and skills, by creating new relations between learners and more knowledgeable others as well as a discourse free space for collaborative learning.

Acknowledgment

This research is conducted in collaboration with the Danish Technological Institute, Robot Technology and Aalborg University, Department of Communication and Psychology.

References

[1] EUROP: "Robotic Visions to 2020 and beyond – The Strategic Research Agenda for robotics in Europe" , 2nd ed., Published by European Robotics Technology Platform, pp. 19, 2009

[2] K. Dautenhahn, et al. "KASPAR – A Minimally Expressive Humanoid Robot for Human-Robot Interaction Research", Applied Bionics And Biomechanics, 3&4, pp. 369-397, 2009

[3] C. J. Lee, K. Kim, C. Breazeal, and R. Picard, "Shybot: Friend-Stranger Interaction for Children Living with Autism", CHI '08 Extended Abstracts on Human Factors in Computing Systems, 2008

[4] G. Majgaard, J. Nielsen, and M. Misfeldt, "Robot Technology and Numbers in the Classroom", Cognition and Exploratory Learning CELDA 2010, International Association for Development, AIDIS pp. 231-234, 2010.

[5] J. Han and K. Dongho, "r-Learning services for elementary school students with a teaching assistant robot", 4th ACM/IEEE International Conference on Human-Robot Interaction, 2009

[6] B. J. Fogg, "Persuasive Computers: Perspectives and Research Directions", SIGCHI Conference on Human Factors in Computing Systems, ACM Press, 1998

[7] B. J. Fogg, "Persuasive Technology – Using Computers to Change What We Think and Do", Morgan Kaufmann, 2003

[8] Bickmore, T. and Cassell, J.: "Relational Agents: A Model and Implementation of Building User Trust", ACM SIGCHI Conference on Human Factors in Computing Systemsm, 2001

[9] S. Vossen, J. Ham, C. Midden, "Social Influence of a Persuasive Agent: the Role of Agent Embodiment and Evaluative Feedback", International Conference on Persuasive Technology, 2009

[10] T. Fong, I. Nourbakhsh, K. Dautenhahn "A Survey of Socially Interactive Robots", Robotics and Autonomous Systems, 42(3-4), p. 143-166, 2003

[11] A. Brockbank, I. McGill, "Facilitating Reflective Learning in Higher Education", 2 ed., Open University Press 1998(2007)

[12] B. F. Skinner, "Contingencies of Reinforcement: A Theoretical Analysis", Upper Saddle River, NJ: Prentice Hall, 1969

[13] M. R. Lepper, D. Greene, R. E. Nisbett, "Undermining Children's Intrinsic Interest with Extrinsic Reward: A Test of the "Overjustification" Hypothesis", Journal of Personality and Social Psychology, 28, 1973

[14] M. Gagné, E. L. Deci, "Self-Determination Theory and Work Motivation", Journal of Organizational Behavior, 26, 2005

[15] J. Piaget, "The Construction of Reality in the Child", Routledge and Kegan Paul Ltd., 1954

[16] M. Hermansen, "Læringens Univers", Klim, 2005

[17] L. S. Vygotsky, "Mind in Society: The Development of Higher Psychological Processes", M. Cole et al. (eds.), Harvard University Press, 1978

[18] D. Feil-Seifer, M. J. Mataric, "Defining Socially Assistive Robotics", 9th International Conference on Rehabilitation Robotics, 2005

[19] P. Øhrstrøm, "Etiske overvejelser og dilemmaer ved brug af velfærdsteknologi i den kommunale social- og sundhedssektor", in "Velfærdsteknologi nye hjælpemidler i ældreplejen", P. Riis (Ed.), Ældreforum, 2010

[20] www.mindstorms.lego.com

[21] M. J. Mataric, N. Koenig, D. Feil-Seifer, "Materials for Enabling Hands-On Robotics in STEM Education", AAAI Spring Symposium on Robots and Robot Venues

[22] G. Majgaard, J. Nielsen, M. Misfeldt, "Robot Technology and Numbers in the Classroom", Cognition and Exploratory Learning CELDA, International Association for Development, AIDIS pp. 231-234, 2011

[23] International Journal of Technology-Enhanced Learning, www.inderscience.com/ijtel

[24] International Journal of Computer Supported Collaborative Learning, www.ijcscl.org

[25] L. Bertel, "The Use of Rewards in Persuasive Design", Poster Proceedings for the 5th International Conference on Persuasive Technology, Oulu University Press, pp. 25-28, 2010

[26] Schärfe, H. and Bertel, L.: "Tracing Concepts in Designing for Change", Proceedings for the 14th World Multi Conference on Systemics, Cybernetics and Informatics, 2010

[27] EuroPLOT: Persuasive Learning Objects and Technologies

[28] Gram-Hansen, S.: "PLOT Persuasive Learning Design Framework: Persuasive Learning Designs", Deliverable D3.3 of the EU-project Persuasive Learning Objects and Technologies for Lifelong Learning in Europe, 2012.

Design of a Passive Exoskeleton for the Upper Extremity through Co-simulation with a Biomechanical Human Arm Model

Lelai Zhou, Shaoping Bai and John Rasmussen
Department of Mechanical and Manufacturing Engineering,
Aalborg University
Fibigerstraede 16, 9220 Aalborg, Denmark
{lzh, shb, jr}@m-tech.aau.dk

***Abstract*— An approach of designing exoskeletons on the basis of simulation of the exoskeleton and a human body model is proposed in this paper. The new approach, addressing the problem of physical human-exoskeleton interactions, models and simulates the mechanics for both the exoskeleton and the human body, which allows designers to analyze and evaluate an exoskeleton for its functioning, effectively. A simulation platform is developed by integrating a biomechanical model of human body and the exoskeleton. With the proposed approach, two types of exoskeletons with gravity compensating capability are designed for assisting patients with neuromuscular injuries. Results of the design analysis and optimization are included.**

Keywords: exoskeleton, human-centered design optimization, biomechanics, physical human-exoskeleton interaction

I. Introduction

Exoskeleton robots have prospective applications in rehabilitation and patients assistance. They can help users to retain independent life by regaining mobility and manipulability.

Exoskeletons can be categorized into two major groups, passive and active exoskeletons. Several passive exoskeleton robots have been developed recently. Wilmington Robotic Exoskeleton (WREX), a two-segment, 4-DOF (degree of freedom) passive orthosis provided by Nemours [1], is a modular body-powered orthosis which can be mounted to a person's wheelchair or to a body jacket. WREX uses linear elastic elements to balance the effects of gravity in three dimensions. A variable impedance powered elbow exoskeleton named NEUROExos [2] was developed for the rehabilitation task of stroke patients. The robot utilizes a double shell link structure and 4-DOF passive mechanism, which has a perfect kinematic compatibility with the user. NEUROExos makes use of an adaptive, passive-compliant actuator through a bio-inspired antagonistic non-linear elastic actuation system. An upper limb exoskeleton with 3-DOF shoulder joint and 1-DOF elbow joint has been designed in [3]. The grounding device can increase resistance through adjusting the spring length to train more muscle groups.

A number of active exoskeletons were also reported recently. The ARMin III [4] is an arm therapy exoskeleton robot with three actuated DOFs for the shoulder and one DOF for the elbow. It was designed to improve the rehabilitation process in stroke patients. The IntelliArm [5] is a whole arm robot, which has a total of eight actuated DOF and another two passive DOF at the shoulder. Besides, the IntelliArm has an additional DOF for hand opening and closing. Several other types of actuated exoskeleton robots were also proposed, such as ABLE [6], CADEN-7 [7], MGA [8], RehabExos [9], and Pneu-WREX [10]. A detail review of the state-of-the-art exoskeleton robots for upper limb could be referred to [11].

It is realized that a successful design of an exoskeleton depends on a better understanding of the biomechanics of the upper extremity motion and sensory mechanisms, which is a critical problem in the physical human-robot interaction. Researchers have tried to model the physical human-robot interaction through musculoskeletal modeling. An attempt was made to model interactions between the human and rehabilitation devices by musculoskeletal simulation [12], where parametric design of devices based on musculoskeletal performance has been conducted.

In this work, a simulation platform is developed for the modeling of the physical human-robot interaction of exoskeletons. The work is conducted based on a preliminary work present in [13]. Our interest is to design a passive exoskeleton with gravity compensation capability. An advanced biomechanical model of the upper extremity is developed by virtue of the AnyBody Modeling System (AnyBody Technology A/S, Aalborg, Denmark), which allows us to understand the mechanics of the bio-robotic system, i.e., the exoskeleton, and finally to design exoskeletons with optimized human-robot interaction.

II. Exoskeleton Design

The exoskeleton utilizes two spring-loaded parallelogram mechanisms to form a serial linkage to support the human arm. Such a design is inspired by the similar mechanisms found in the table lamp stands.

The passive exoskeleton utilizes two parallelogram mechanisms with springs to compensate the gravity, as shown in Fig. 1. In each parallelogram mechanism, the spring is set co-linearly with the lower bar to save the space for the spring. The other end of the spring is anchored at the vertical bar, as A_1 or A_2 in Fig. 1(b), while passing through B_1 or B_2. The springs are installed so that when $\theta_{e1} = \pi/2$ or $\theta_{e2} = \pi$, the springs have their natural lengths. The distances $\overline{A_1B_1}$ and $\overline{A_2B_2}$ are the elongation of the upper and lower spring, respectively.

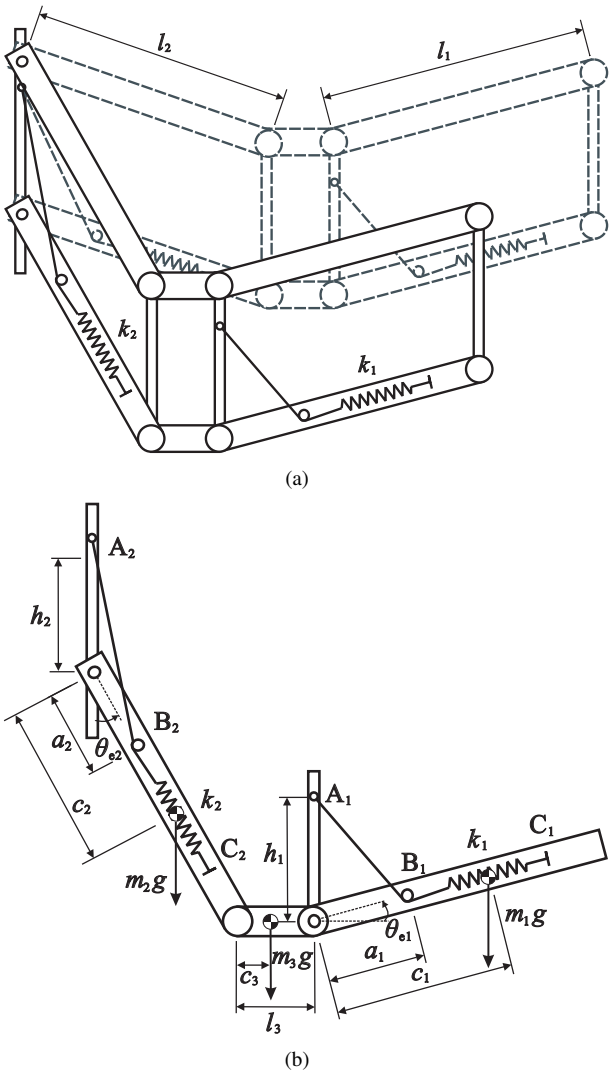


Fig. 1. Gravity compensation system of the exoskeleton, (a) parallel structure, (b) schematic diagram of the spring system.

When the lower parallelogram mechanism is balanced, the torque generated by the gravity force shall be balanced by the spring force. The equilibrium equation of the mechanism can be expressed as

$$m_1 g c_1 \cos(\theta_{e1}) - k_1 a_1 h_1 \sin(\pi/2 - \theta_{e1}) = 0 \qquad (1)$$

The stiffness coefficient of the spring in the lower parallelogram mechanism can be solved as

$$k_1 = \frac{m_1 g c_1}{a_1 h_1} \qquad (2)$$

where m_1 is the mass of the lower link, c_1 locates the center of mass, h_1 and a_1 are the distances from the anchoring and wrapping point to the joint, as illustrated in Fig. 1. According to Eq.(2), the spring stiffness k_1 required for the equilibrium condition is independent to the joint angle θ_{e1}. A spring with constant stiffness can be selected for the lower parallelogram mechanism.

When the upper link is balanced, the moment generated by the weight is calculated as

$$\begin{aligned} T_G &= m_1 g(c_1 \cos(\theta_{e1}) + l_2 \sin(\theta_{e2}) + l_3) \\ &\quad + m_2 g c_2 \sin(\theta_{e2}) + m_3 g(c_3 + l_2 \sin(\theta_{e2})) \end{aligned} \qquad (3)$$

where m_2 and m_3 are the mass of the upper parallelogram mechanism and middle link respectively. l_2 and l_3 are lengths of the upper and middle links. c_2 and c_3 locate the centers of mass of the upper and middle links. The moment generated by the spring force can be obtained as

$$T_S = k_2 a_2 h_2 \sin(\theta_{e2}) \qquad (4)$$

Given that $T_G - T_S = 0$, the stiffness of the spring is solved as

$$\begin{aligned} k_2 &= \frac{m_1 g(c_1 \cos(\theta_{e1}) + l_3) + m_3 g c_3}{a_2 h_2 \sin(\theta_{e2})} \\ &\quad + \frac{m_1 g l_2 + m_2 g c_2 + m_3 g l_2}{a_2 h_2} \end{aligned} \qquad (5)$$

The spring in the upper link requires a variable stiffness in order to statically compensate the gravity at any point, as the stiffness co-efficient is related to the joint angles θ_{e1} and θ_{e2}. If the parameters in Fig. 1(b) are given as $m_1 = 2kg$, $m_2 = 3kg$, $m_3 = 1kg$, $c_1 = c_2 = 0.2m$, $c_3 = 0.02m$, $l_2 = 0.26m$, $l_3 = 0.04m$, $h_1 = h_2 = a_1 = a_2 = 0.04m$, the stiffness of the spring has a profile as in Fig. 2.

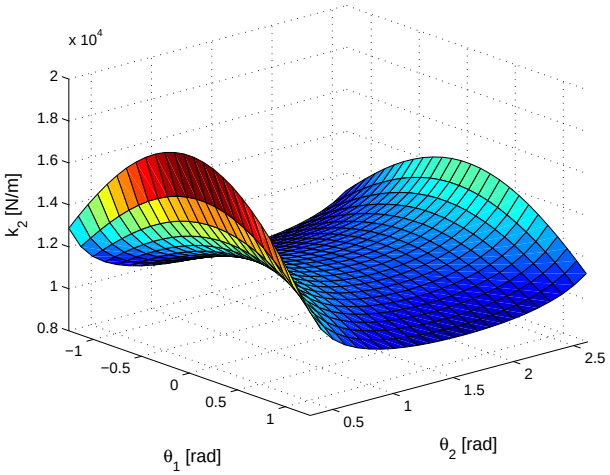


Fig. 2. Contour plot of spring stiffness.

III. Modeling of the Bio-exoskeleton System

A bio-exoskeleton system refers in this work to an exoskeleton working cooperatively with human muscles and nerves. The interaction between the exoskeleton and the human body determines whether the exoskeleton can implement the desired functions. A central issue in the modeling work is thus to simulate the response of human body subject to external force/torques exerted by the exoskeleton.

The bio-exoskeleton system model consists two modules: a musculoskeletal model and an exoskeleton model. The musculoskeletal model deals with biomechanics analysis, and the exoskeleton model conducts the kinematics and dynamics simulation.

A. Biomechanical Modeling

In a musculoskeletal model, the human body is modeled as a multibody system, in which bones and joints are treated as mechanical links and joints, while muscles exert force on the system. It is known that the system is statically indeterminate, the muscle recruitment can be formulated as an optimization problem as

$$\begin{aligned} \min \quad & G(\mathbf{f}^{(M)}) \\ \text{s.t.} \quad & \mathbf{Cf} = \mathbf{d} \\ & f_i^{(M)} \geq 0, \quad i \in \{1, \ldots, n^{(M)}\} \end{aligned} \tag{6}$$

where $\mathbf{f} = [\mathbf{f}^{(R)}, \mathbf{f}^{(M)}]$ is composed of a n-dimensional vector of joint reaction forces $\mathbf{f}^{(R)}$ and muscle forces $\mathbf{f}^{(M)}$. The vector $\mathbf{d}$ is the external force, and $\mathbf{C}$ is a coefficient matrix generated from the arm anatomy and muscle attachments. The choice of the objective function $G(\mathbf{f}^{(M)})$ depends on the muscle recruitment criterion. The possible criteria include soft saturation, min/max [14] and polynomial muscle recruitment, etc. The polynomial criterion is adopted as

$$G(\mathbf{f}^{(M)}) = \sum_i \left(\frac{f_i^{(M)}}{N_i} \right)^p \tag{7}$$

where N_i are normalization factors or functions, which take the form of muscle strength in this work. The power p indicates the synergy of muscles. $p = 3$ is recommended as it yields good results for most submaximal muscle efforts. The ratio $f_i^{(M)}/N_i$ refers to the muscle activity.

In certain cases of simulations with paralyzed muscles, the model needs to disable any single piece of muscle. In the model, a muscle is disabled by setting its isometric strength to zero. When several muscles are disabled, their forces are correspondingly set to zero. In return, the equilibrium equation $\mathbf{Cf} = \mathbf{d}$ in Eq. (6) implies that the other muscles must be recruited differently to balance the external load. When too many muscles are paralyzed, to make sure that the muscle recruitment always has a solution, weak artificial muscles are included in the model.

B. Bio-exoskeleton Model

The bio-exoskeleton is developed through integrating the human arm model and the exoskeleton. A musculoskeletal model of the right arm was built in the AnyBody Modeling System. The whole musculoskeletal model is comprised of 39 joints and 134 muscles. The model is derived from the repository models in AnyBody.

The exoskeleton model was built in SolidWorks and then exported to AnyBody. Several reference nodes were defined on the human model for placing the armor and brackets of the exoskeleton. The mass properties of the exoskeleton can be calculated in SolidWorks, and redefined in AnyBody afterwards. All joints need to be defined in AnyBody. The attachment of the lower parallelogram mechanism to the wrist was modelled as a spherical joint.

The force in the spring is defined as

$$F_s = k * \Delta l \tag{8}$$

where k is the spring stiffness and Δl is the variation of the spring length.

When integrating the exoskeleton with the arm model in AnyBody, the muscle recruitment in Eq. (6) is transformed as

$$\begin{aligned} \min \quad & G(\mathbf{f}^{(M)}) \\ \text{s.t.} \quad & \mathbf{C}^*\mathbf{f} = \mathbf{d}^* \end{aligned} \tag{9}$$

where the external force $\mathbf{d}^*$ contains the force generated by the exoskeleton. The exoskeleton attaching to the arm also transforms the coefficient matrix $\mathbf{C}^*$.

IV. Simulations

Simulations were conducted using the model developed. The motion of the musculoskeletal model uses the motion capture data of people's activities of daily living.

A. Arm Motion

The motion data is used to drive the musculoskeletal human arm model for kinematics analysis. In this work, a customized motion capture system is built by two Kinect^{TM} sensors. A motion of picking up a cup and drinking is captured within 3 seconds.

The arm trajectory of the captured motion is depicted in Fig. 3. In the captured motion, the movement of the two wrist joints and the elbow pronation were ignored. Only the other four joints of the human arm are actuated in the simulation.

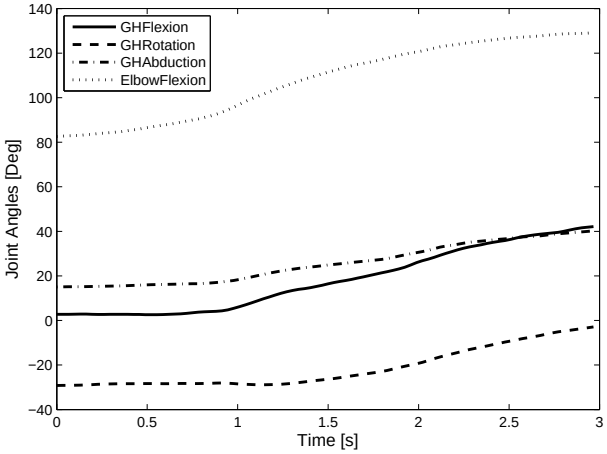


Fig. 3. Arm joint rotation in the motion of picking up a cup. (GH denotes the glenohumeral joint.)

B. Exoskeleton Configurations

Two types of exoskeletons, i.e. Type I and II, are proposed, as shown in Fig. 4 and Fig. 5. The only difference between the Type-I and Type-II is the direction of joint 1 (θ_1). In the Type-I, the joints 1, 2 and 3 form a spherical joint. If the upper arm abducts, the joint 1 rotates to follow the motion such that the two parallelogram links could not be always in vertical planes.

In the Type-II, the axis of the joint 1 is set parallel to joints 2 and 4. In the design, the whole system can be always

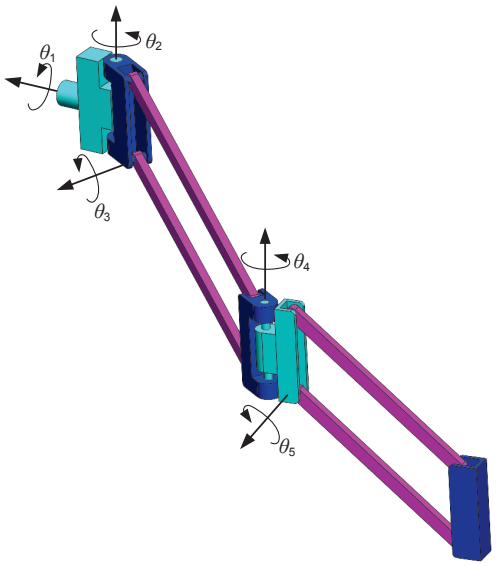


Fig. 4. The kinematic model of the exoskeleton (Type-I).

kept perpendicular to the ground. The exoskeleton can fully exhaust its capability to compensate the gravity in the vertical direction. The Type-II has a drawback as if the upper arm abducts externally, the exoskeleton has to leave some space to the arm in case of collision. Whereas, the Type-I can be attached to the human arm closely.

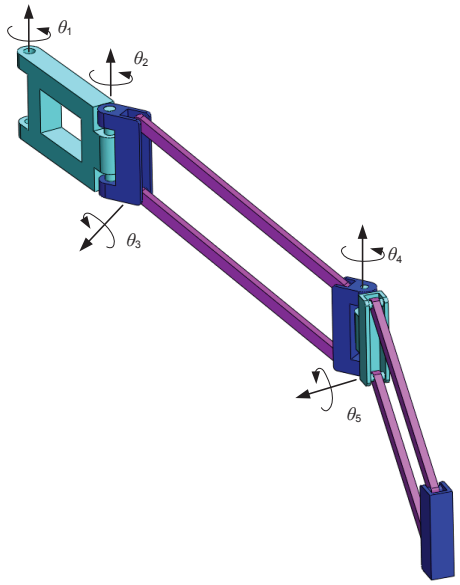
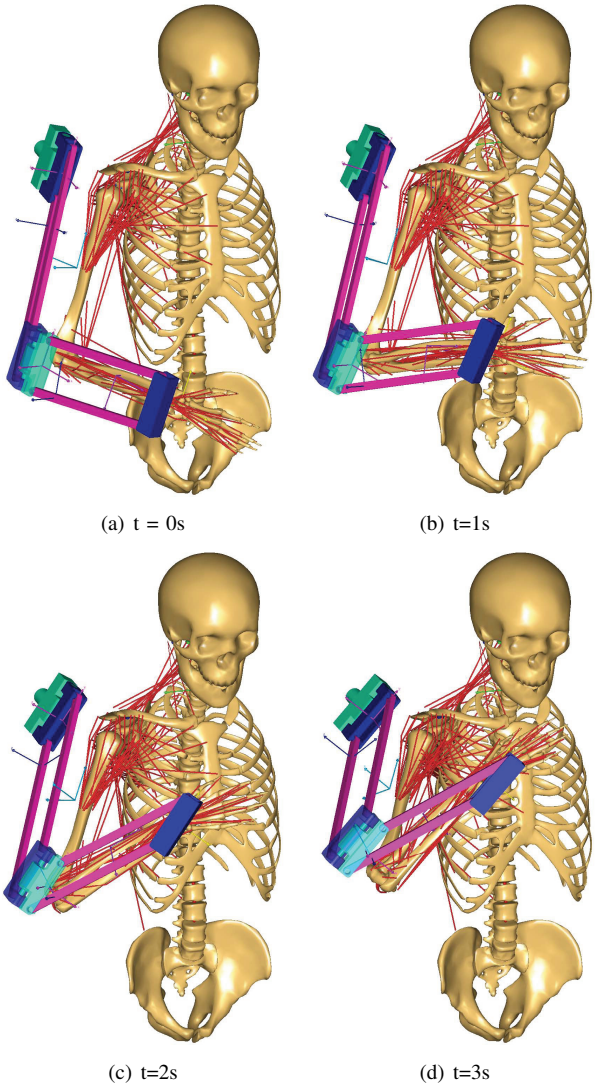


Fig. 5. The kinematic model of the exoskeleton (Type-II).

C. Exoskeleton Kinematics

With the same human arm motion, the solved exoskeleton kinematics for the Type-I and Type-II are different, as shown in Figs. 6 and 7. The kinematic simulation shows that the Type-II exoskeleton needs more space to the human arm for effective moving.

The solved trajectories of the Type-I and Type-II exoskeletons are shown in Fig. 8 and Fig. 9, respectively.



(a) t = 0s (b) t=1s

(c) t=2s (d) t=3s

Fig. 6. Motion of picking up a cup (Type-I)

D. Design Case

A medical case called brachial plexus injury is studied in this work. The brachial plexus is a network of nerves that conducts signals from the spinal cord, which is housed in the spinal canal of the vertebral column (or spine), to the shoulder, arm and hand. These nerves originate in the fifth, sixth, seventh and eighth cervical (C5-C8), and first thoracic (T1) spinal nerves, as demonstrated in Fig. 10(a). They innervate the muscles and skin of the chest, shoulder, arm and hand. Injuries of brachial plexus, or lesions, often caused by trauma conditions such as traffic accidents can have serious effect on the mobility of limbs [15].

The branches of the brachial plexus and their associated muscles are listed in Table I, sorted with respect to their roots. The spinal nerves and their cord related to brachial plexus are shown in Fig. 10(a) and Fig. 10(b).

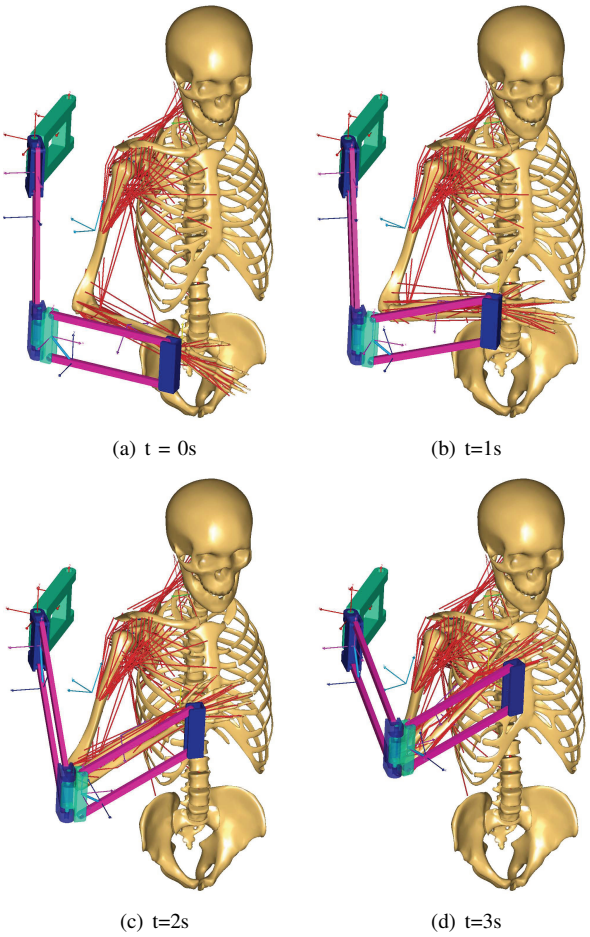
E. Optimization on the Exoskeleton

In this work, the design of the exoskeleton is formed as an optimization problem through biomechanical simulation. The

TABLE I

BRANCHES OF BRACHIAL PLEXUS.

No.	Nerve	Roots	From	Muscles
1	axillary	C5, C6	posterior cord	deltoid, teres minor
2	radial	C5, C6, C7, C8, T1	posterior cord	triceps brachii, supinator, anconeus, the extensor muscles of the forearm, brachioradialis
3	upper subscapular nerve	C5, C6	posterior cord	subscapularis(upper part 1 2)
4	lower subscapular nerve	C5, C6	posterior cord	subscapularis(lower part 3 4 5 6), teres major
5	thoracodorsal nerve	C6, C7, C8	posterior cord	latissimus dorsi
6	lateral pectoral nerve	C5, C6, C7	lateral cord	pectoralis major
7	musculocutaneous nerve	C5, C6, C7	lateral cord	coracobrachialis, brachialis, biceps brachii
8	suprascapular nerve	C5, C6	upper trunk	supraspinatus, infraspinatus
9	long thoracic nerve	C5, C6, C7	root	serratus anterior
10	dorsal scapular nerve	C5	root	rhomboid muscles, levator scapulae



(a) t = 0s (b) t=1s

(c) t=2s (d) t=3s

Fig. 7. Motion of picking up a cup (Type-II)

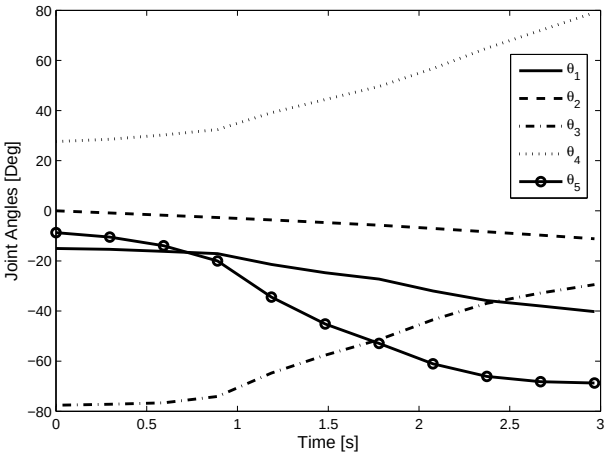


Fig. 8. Type-I exoskeleton joint angles in the motion of picking up a cup.

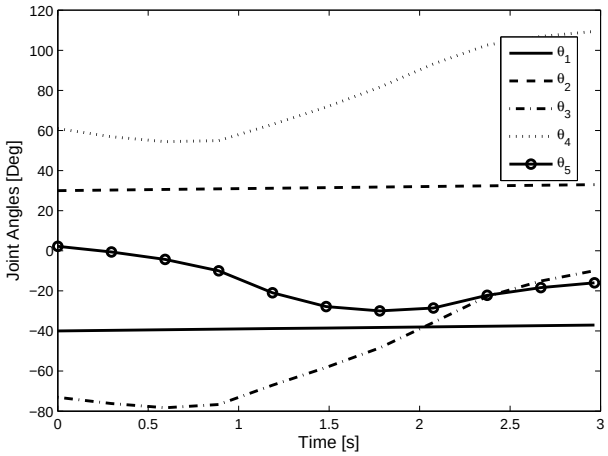
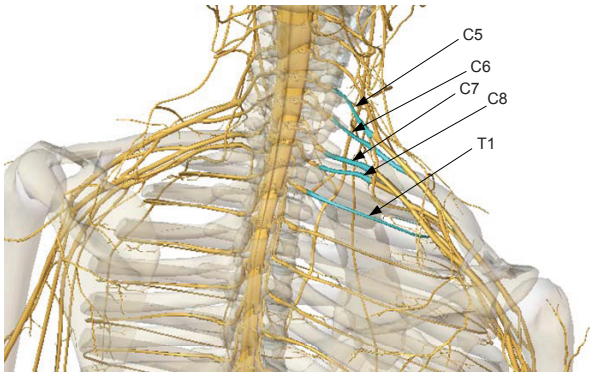


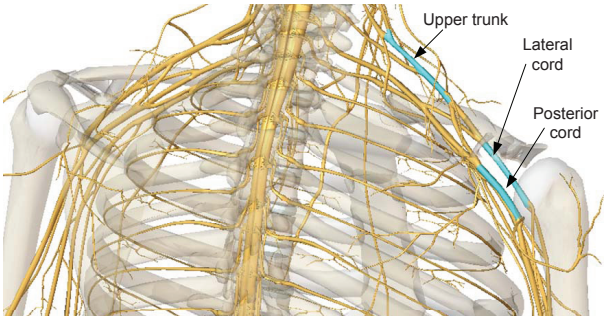
Fig. 9. Type-II exoskeleton joint angles in the motion of picking up a cup.

stiffness coefficient of the spring in the upper parallelogram link is defined as k_u. The stiffness of the lower spring is k_l. In the biomechanics simulation, the muscle recruitment criterion in Eq. (6) and (7) is to minimize the polynomial function of muscle activation with $p = 3$. When set $p = 1$ to Eq. (7), we can get the sum of all the muscles activation. The objective of the exoskeleton is to reduce the maximal muscle activation (MMA). The objective function is defined as

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = \max\left\{\sum_i \frac{f_i^{(M)}}{N_i}\right\} \\ & \mathbf{x} = [k_u, k_l] \\ \text{s.t.} \quad & \min \; G(\mathbf{f}^{(M)}) \\ & \mathbf{C}^*\mathbf{f} = \mathbf{d}^* \end{aligned} \tag{10}$$



(a)



(b)

Fig. 10. The spinal nerve cords of the nervous system.

There are two design variables in the optimization problem. The optimization problem is solved by the Complex method [16], [17]. This optimization problem is wrapped around the full inverse dynamic analysis of the model. In each iteration, the maximal MMA is calculated over the motion duration after inverse dynamics is completed.

V. Results and Discussions

A. Maximal Muscle Activation

The muscle activity is a fraction of maximum voluntary contraction (MVC). When it exceeds 1, the arm does not have the strength to complete the required motion. To allow the simulation to complete in cases of inadequate muscle strength, weak artificial muscles have been added to the joints, thus allowing all the cases of lesions to be simulated, albeit in some cases with very high activation levels.

The maximal muscle activation is calculated for different nerve lesion conditions. We categorized the muscles into groups according to the nerve root and origin, as shown in Table I. For example, if the nerve root C5 has a lesion, all the muscles list in the table except No.5 will be paralyzed. The payload at hand is $0.5kg$. The calculated maximal MMA of different nerve lesion is shown in Fig. 11. Note that the case BASE refers to the motion without any nerve lesion.

It is found that paralyzing nerve C5 or C6 will lead to very high MMA. It is reasonable as lesion on C5 or C6 paralyzes most of the functional muscles in the upper arm and shoulder. The highly required muscle activity indicates that if there is lesion at C5 or C6, it might be hard to design

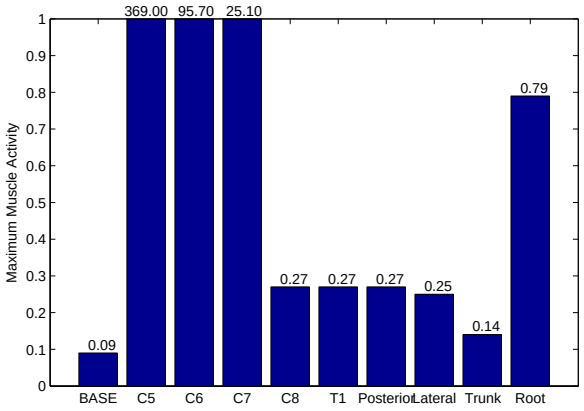


Fig. 11. Maximal MMA of different nerve lesion cases.

a pure passive orthosis for the arm. The maximal MMA of nerve lesions C8, T1, Posterior cord, Lateral cord, Upper trunk and Root does not exceed 1, such that these nerve lesions do not need the assistance of the exoskeleton. The case C7 with a maximal MMA of 25.1 is then selected as the patient case of the exoskeleton.

B. Optimization Results

Optimization is executed on the Type-II exoskeleton. For the case C7 in the motion of picking up a cup, we set a population number of 10 to the Complex method for executing optimization. The objective convergence tolerance is 0.001, and the convergence tolerance for the design variables is 0.001. The maximal muscle activation is reduced from 25.1 to 0.54 after optimization with 202 iterations. The convergence of the maximal muscle activation is plotted in Fig. 12.

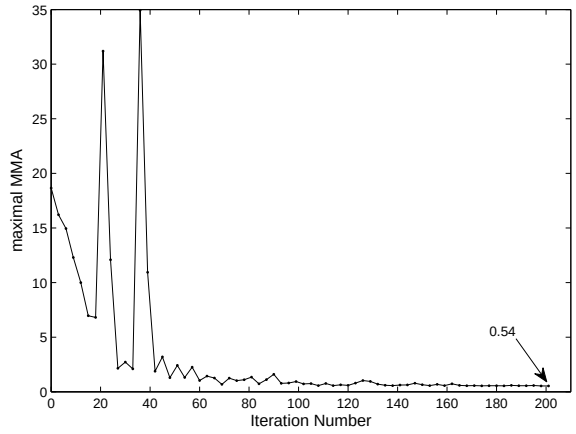


Fig. 12. Optimization of the maximal MMA.

The optimal design variables are obtained as

$$k_u = 5423\ N/m$$
$$k_l = 2774\ N/m \quad (11)$$

The converging history of stiffness co-efficient is shown in Fig. 13. The optimization results provide useful information for the optimal design of the exoskeleton.

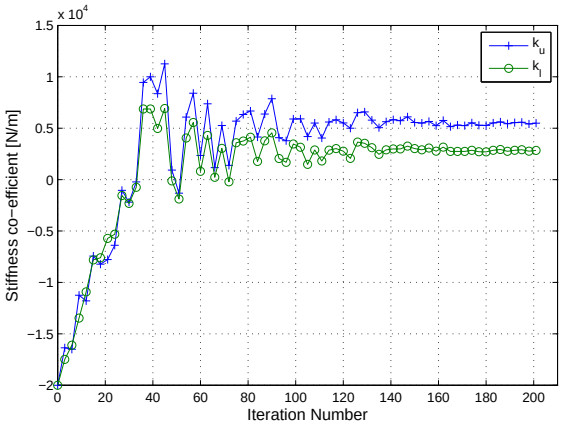


Fig. 13. Convergence of the stiffness co-efficient.

With the optimal parameters, the MMA is calculated for the case without exoskeleton and the one with the optimal design, as plotted in Fig. 14. Wearing the optimal designed exoskeleton, the MMA of the arm is always below 1 in the motion. The patient can perform the activity with the proposed exoskeleton.

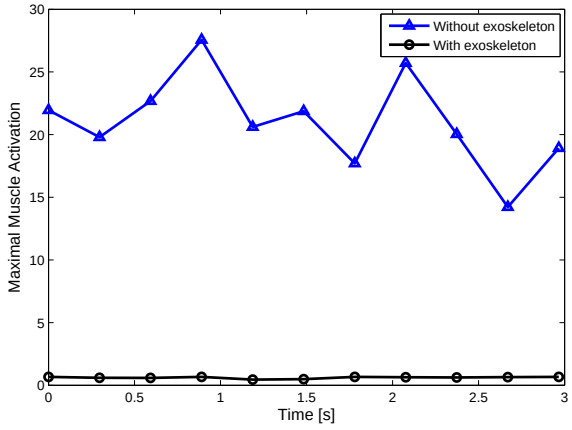


Fig. 14. The comparison of the MMA with and without exoskeleton assistance.

VI. Conclusions

An approach of designing exoskeletons through biomechanics simulation was proposed. The approach integrated the exoskeleton model with a biomechanical arm model to optimally design the exoskeleton. Two types of pure passive exoskeletons were proposed and simulated with the developed model. The exoskeleton was optimized based on evaluation of the performance of the musculoskeletal human arm model. The prototype will be built to testify the model and validate its application in patients' daily living.

Acknowledgement

The work is supported by the strategic platform for research and innovation project, "Patient @ home", funded by the Danish Agency for Science, Technology and Innovation.

References

[1] T. Rahman et al. Passive exoskeletons for assisting limb movement. *Journal of Rehabilitation Research & Development*, 43(5):583–590, 2006.

[2] T. Lenzi, N. Vitiello, S. M. M. De Rossi, S. Roccella, F. Vecchi, and M. C. Carrozza. NEUROExos: a variable impedance powered elbow exoskeleton. In *Proc. of IEEE Inter. Conf. on Robotics and Automation*, pages 1419–1426, Shanghai, China, 2011.

[3] T. M. Wu, S. Y. Wang, and D. Z. Chen. Design of an exoskeleton for strengthening the upper limb muscle for overextension injury prevention. *Mechanism and Machine Theory*, 46(12):1825–1839, 2011.

[4] T. Nef, M. Guidali, and R. Riener. ARMinIII - arm therapy exoskeleton with an ergonomic shoulder actuation. *Applied Bionics and Biomechanics*, 6(2):127–142, 2009.

[5] Y. Ren, H. S. Park, and L. Q. Zhang. Developing a whole-arm exoskeleton robot with hand opening and closing mechanism for upper limb stroke rehabilitation. In *Proc. of IEEE Inter. Conf. on Rehabilitation Robotics*, pages 761–765, 2009.

[6] P. Garrec, J. P. Friconneau, Y. Measson, and Y. Perrot. ABLE, an innovative transparent exoskeleton for the upper-limb. In *Proc. of IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems*, pages 1483–1488, 2008.

[7] J. C. Perry, J. Rosen, and S. Burns. Upper-limb powered exoskeleton design. *IEEE/ASME Transactions on Mechatronics*, 12(4):408–417, 2007.

[8] C. Carignan, J. Tang, and S. Roderick. Development of an exoskeleton haptic interface for virtual task training. In *Proc. of IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems*, pages 3697–3702, 2009.

[9] R. Vertechy, A. Frisoli, A. Dettori, M. Solazzi, and M. Bergamasco. Development of a new exoskeleton for upper limb rehabilitation. In *Proc. of IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems*, pages 188–193, 2009.

[10] E. T. Wolbrecht, V. Chan, D. J. Reinkensmeyer, and J. E. Bobrow. Optimizing compliant, model-based robotic assistance to promote neurorehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 16(3):286–297, 2008.

[11] H. S. Lo and S. Q. Xie. Exoskeleton robots for upper-limb rehabilitation: State of the art and future prospects. *Medical Engineering & Physics*, 34(3):261–268, 2012.

[12] L. F. Lee, M. S. Narayanan, S. Kannan, F. Mendel, and V. N. Krovi. Case studies of musculoskeletal-simulation-based rehabilitation program evaluation. *IEEE Transactions on Robotics*, 25(3):634–638, 2009.

[13] S. Bai and J. Rasmussen. Modelling of physical human-robot interaction for exoskeleton designs. In J. C. Samin and P. Fisette, editors, *Proc. of Multibody Dynamics 2011, ECCOMAS Thematic Conference*, Brussels, Belgium, 2011.

[14] J. Rasmussen, M. Damsgaard, and M. Voigt. Muscle recruitment by the min/max criterion: A comparative numerical study. *Journal of Biomechanics*, 34(3):409–415, 2001.

[15] C. J. Standaert and S. A. Herring. Expert opinion and controversies in musculoskeletal and sports medicine: stingers. *Archives of Physical Medicine and Rehabilitation*, 90(3):402–406, 2009.

[16] M. J. Box. A new method of constrained optimization and a comparison with other methods. *Computer Journal*, (8):42–52, 1965.

[17] J. A. Guin. Modification of the complex method of constrained optimization. *Computer Journal*, 10:416–417, 1968.

Human Assisted Computer Vision on Industrial Mobile Robots

Rasmus S. Andersen * **Casper Schou** * **Jens S. Damgaard** *
Ole Madsen * **Thomas B. Moeslund** **

* *Dep. of Mechanical and Manufacturing Engineering, Aalborg University, Denmark*
** *Dep. of Architecture, Design and Media Technology, Aalborg University, Denmark*

ABSTRACT

Much research is directed at developing increasingly efficient and flexible production, and one important potential advancement is *Autonomous Industrial Mobile Manipulators* (AIMM's). The idea behind AIMM's is to have robots that have the ability to perform a wide variety of tasks, and which can easily and efficiently be reconfigured when the requirements changes. In this paper, the paradigm of *skill based* programming is investigated, and in particular how computer vision abilities can be integrated in this. Three applications of computer vision developed in a skill based framework are presented; namely vision pick, quality control, and fast calibration. All three are implemented on Aalborg University's AIMM, *Little Helper*, and tested in a real-life industrial environment at the Danish company Grundfos A/S.

1. INTRODUCTION

The globalization has for several decades moved manufacturing jobs from western countries to low-wage developing countries. This has put pressure on both wages and the productivity of production in the industrialized countries. One efficient way of increasing productivity is to automate production by using robots. A major limitation for the application of robots is, however, the scale of production. Construction of an automated production line is a major investment, and configuration of robots to perform the required operations is a time consuming task, that must be performed by highly specialized engineers. Thus, installation of a new, fully automated production line can only be justified if the quantity of identical items to be produced is very large. Robots have therefore proven to be particularly useful in industries such as in the car manufacturing industry, where a large quantity of identical products have to be produced.

For many kinds of production, the amount of identical items is, however, not large enough to justify investment in automated robotic production lines. Much research have therefore been directed towards developing more flexible types of automation. The organization *European Robotics Technology Platform* (EUROP) published in 2009 a *Strategic Research Agenda for European Robotics* (SRA), which outlined areas that European robotics research should focus on as well as metrics for each area EUROP (2009a,b). The core requirements for future robotics include:

- Reconfigurability: It must be possible to reconfigure both robots and other production hardware fast and easy, to prevent expensive idle time for long periods between production of (possibly small) production series.
- Human Robot Interaction: The communication between robot and human operators must be intuitive and to an increasing degree use languages and interfaces that are natural to humans.
- Autonomy (ability to function in less structured environments): On a car manufacturing plant, robots work in highly structured environments, virtually without any human presence at all. This approach is not sufficient for smaller production series. Thus, the robots must have a larger degree of autonomy, enabling them to perform tasks in dynamic environments, that cannot be precisely modeled before production begins.

One type of industrial robot that is well suited for such a production scenario, is the *Autonomous Industrial Mobile Manipulator* (AIMM). Although AIMM's are not yet in industrial use, they are already able to move autonomously around in changing environment and perform a wide variety of tasks. Since AIMM's must be designed to function in less than fully structured environments, they depend very much on their ability to sense both objects and the world around them. The focus of this paper is to investigate methods to do such sensing by using computer vision in a fast and easily reconfigurable way.

1.1 Related Research

Ordinary RGB cameras are used to give vision functionality to robots in various fields, including navigation, object manipulation, and interaction with humans, cf. Hvilshøj et al. (2009); Guizzo and Ackerman (2012); Nava et al. (2011). The human visual system includes additionally information about depth, and several approaches have been taken to provide this information to robots also, including stereo vision (Murray and Little, 2000), time-of-flight (TOF) depth cameras (Klank et al., 2009), and

* This research was partially funded by the European Union project TAPAS under the Seventh Framework Programme.

depth cameras based on structured light (Siegwart and Nourbakhsh, 2004). While stereo vision is the closest analogue to the human visual system, it is far simpler to use active technologies such as TOF or structured light. With the launch of the Microsoft Kinect in 2010, which combines an ordinary RGB camera with a depth camera based on structured light, the price and accessibility of quality depth video imaging was all of a sudden reduced dramatically (Shotton et al., 2011; El-laithy et al., 2012). This dramatically increased the scientific interest in taking advantage of depth information in combination with RGB images for all areas, where RGB images was also previously used Tölgyessy and Hubinskỳ (2010); Benavidez and Jamshidi (2011); León et al. (2011). Recently, the smaller but equally powerful competitor *Asus Xtion Pro Live* was launched.

Since AIMM's must have the ability to move between workstation, calibration to new workstations is a particular useful aspect, which has also received some attention in the literature. In 2000, a general method for camera calibration was developed by Zhang (2000). This has later become extremely popular, due to implementations provided both for C/C++ in OpenCV and for Matlab in the Matlab camera calibration toolbox. This is designed specifically for estimating parameters, intrinsic as well as extrinsic, for cameras, and not directly applicable for calibration of robots. Another approach by Alici and Shirinzadeh (2005) calibrates industrial robots with very high precision, but this require a laser tracker to be located close to the calibration point. Thus, it is not suitable for AIMM's, which should be able to work in industrial environments without requiring extensive and/or expensive changes.

Two approaches from Hvilshøj et al. (2010) are specifically developed to AIMM's. A fast approach use in addition to a camera a laser for distance measurements, and a slower but very precise method makes only use of a camera on the tool. Both methods have, however, disadvantages: The fast approach requires that a laser is mounted on the tool of the robot. More equipment on the tool means less possible payload, and must therefore be avoided if possible. In the more precise approach, a large number of images are captured of a calibration board, and the execution time is about 60 seconds. If the robot is moving frequently between workstations, such non-productive time must be minimized.

A last approach, described in Pedersen (2011), applies haptic rather than vision based calibration. This approach is a able to calibrate very precisely in about 30 seconds by measuring locations on the workstation in three orthogonal directions. The disadvantage with this method is, in addition to the relatively long execution time, that the workstation must be have large surfaces in all three directions. Also, it is only applicable on robots with force feedback control.

1.2 Skill Based Computer Vision

A traditional and widely used way of programming robots is the *Sense-Plan-Act* (SPA) paradigm (Nilsson, 1993). Using this, the robot moves between the three states: Sense, plan and act. In the sensing state, information from sensors are used to update and maintain a world model. In the planning state, high level logic plans on basis of this world model what the robot has to do, and in the acting state the plan is carried out, typically using control theory. Two limitations of the SPA paradigm is that it does not well support reusability of code, and that the complexity of maintaining a complete world model can be very high.

The paradigm of robot skills attempts to counter both limitations of the SPA paradigm by introducing a layered architecture, where each layer executes its own SPA loop. The idea of using layers to provide better possibilities for reusing code was presented as early as in 1986 by Brooks (1986), but research to provide even more reusable and more generic solutions continue, cf. Gat (1998); Bjorkelund et al. (2011). In the skill paradigm, programming is divided in three layers. Different naming conventions exist, and here the layers are named *device primitives*, *skills*, and *tasks*. The purpose of the layered programming structure is to wrap the difficult and low level robotic knowledge in the lower levels, allowing non-expert users to focus on teaching tasks on a much higher level.

In the ongoing research project 'Little Helper' at Aalborg University, AIMM's have been developed on the basis of the skill paradigm since 2008. In close collaboration with both academic and industrial partners, it is attempted to make the technology ready for industrial use.

In this paper, the integration of computer vision in the skill based framework is presented. The vision algorithms are implemented using a commercial computer vision system based on *Labview* as well as the open source library *zbar*, and the focus here is on how to integrate and use this in the skill based robotic framework. First, the skill paradigm and the vision system applied are described in detail. Subsequently, three developed applications of computer vision are described: A generic pick skill using vision, quality control integration, and a fast calibration based on recognizing QR codes. Finally it is discussed how the results can be generalized, and where future research should be directed. The results presented in the paper are from a midway demonstration in the EU project TAPAS, performed in a factory owned by the Danish company Grundfos A/S.

2. METHODS

2.1 The Concept of Robot Skills

The architecture in the skill paradigm used here consists of three layers:

(1) **Device primitives:** Basic functions of one device, such as the robot, tool or a camera. Example: *Open gripper*.
(2) **Skills:** A predefined sequence of device primitives, that form a coherent action. In Björkelund et al. (2011), a skill is defined as "productive sensor-based robot motions". Example: *Pick up object* O_1.
(3) **Tasks:** Responsible for achieving the overall goals of the robot, while at the same time completely decoupled from the internals of the robot. The robot itself can thus in principle be replaced without replacing the task layer, as long as the new robot provides the same skills. Example: *Pick up 10 units of object* O_1 *at location* L_1*, and place them in a bin at location* L_2.

It does only make sense to execute a place skill, if the robot is holding an object. This means that a precondition for a place skill is, that an object is held. In general a skill has both pre- and post-conditions, and the skill only functions if these are met. This property is in general called that skills are *situated*.

In the skill paradigm, each skill has a *teach* and an *execute* phase. If the user wants the robot to pick up an object of type O_1 from location L_1, a pick is chosen and taught. After teaching is completed, the robot is able to execute the same skill, thus picking a new object of the same type from location L_1 on its own. Prerequisites include here include that the robot is already located at (or close to) location L_1, and that an object of type O_1 is present at the location.

2.2 Flexible Setup with External Computer Vision System

The computer vision system used here is based on the Vision Builder software in Labview. It is able to perform a large number of 2D vision tests, and it employs an intuitive interface, allowing non-experts to configure it with very little training. A screen image is shown in Figure 1.

This paper is, however, not concerned with the vision system itself, but with the integration and use of vision systems in general in the skill based robotic framework. The vision system is not installed on the robot itself. Instead, a protocol based on TCP/IP has been developed, which allows the robot to communicate with an external vision system. To be able to integrate the robot in different kinds of production lines, support for both local camera (mounted on the robot) as well as external cameras has been included in the protocol. For the experiments using the vision system that are described here, two cameras are used; one placed on the tool of the robot, and one fixed at an assembly cell. Both cameras are of the type DMK 31BF03-Z2, which have motorized zoom and a resolution of 1024×768. The first camera is used to detect and pose estimate rotor caps to be able to pick them up, while the fixed camera is used to perform quality control during assembly. Each of these cases are presented in the following subsections, along with the remaining experiment; providing fast calibration by pose estimation of QR codes.

2.3 Generic Vision Pick

The developed vision pick skill consists like all skills of a training and an execution phase. In the training phase, the following parameters are taught:

Fig. 1. Vision detection system in execution mode.

Start and end position of camera: These positions define both the route the camera will take when searching for an object to pick up, as well as an *acceptance region* for objects. During execution, the robot will move the camera from the start towards the end position. With short intervals, the robot will stop and grab an image in search of an object to pick up. Whenever an object has been found, it is calculated if the object is located in the acceptance region; between the two points. If this is the case, the robot cancels the movement towards the endpoint and picks up the object instead.

Detection height: Height of the feature, that the vision system is able to detect.

Grasping height: Appropriate height for grasping the object.

All the required parameters are taught by manually moving the robot arm around, and thus no programming skills are required. The parameters are illustrated in Figure 2. The principle for execution is to first capture an image, and then try to detect the location of a particular feature on the object to pick up in this image. This 2D position can be transformed into a 3D vector from the camera's focal point to the image plane, by applying the intrinsic parameters of the camera. By extending this vector, it will ultimately intersect the plane with the (taught) detection height, as illustrated in Figure 2. If this 3D intersection point is located in the acceptance region, the object can be picked. An additional height; the grasping height, is taught during training, and this enables the robot to grasp the object on a suitable position. The algorithm for execution of the vision pick skill is described in detail in Table 1.

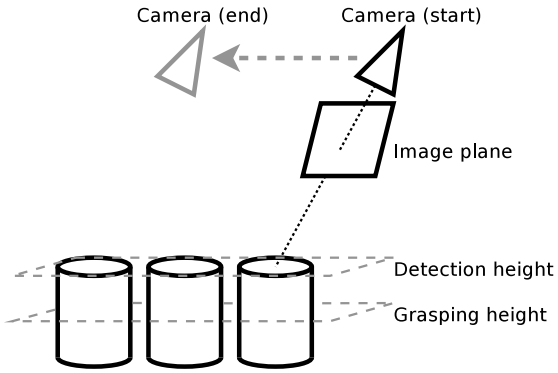


Fig. 2. Execution of the vision pick algorithm require the parameters $\boldsymbol{p}_{\text{cam,start}}$, $\boldsymbol{p}_{\text{cam,end}}$, h_{detect}, and h_{grasp} which are shown in the figure. All the parameters are specified during teaching.

The setup is shown in Figure 3 for two different locations. The Figures 3(a) and 3(c) show the robot searching for rotor caps, while Figure 3(b) and 3(d) show the robot actually picking up a robot cap. Note that the same skill is used at the different locations; only the parameters that are set during teaching differ.

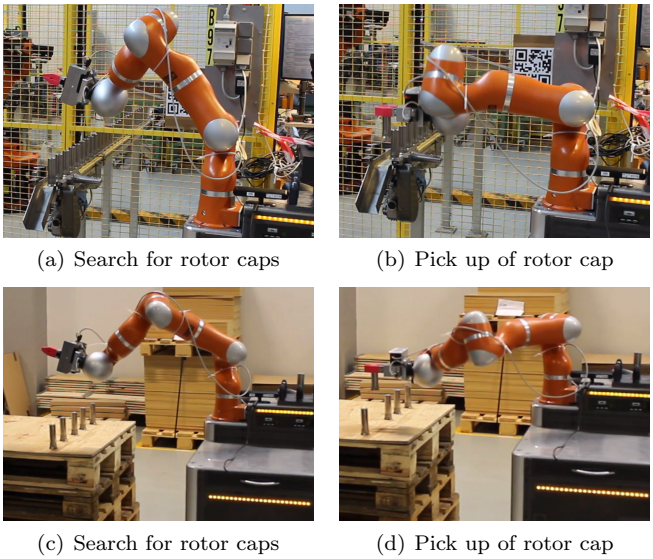
2.4 Quality Control

As with the vision pick skill, also the quality control is developed by utilizing the vision system described in Section 2.2. Thus, both cameras on the robot as well as external cameras can be used. Configuring quality control

(1) Move camera to (taught) start position, $\boldsymbol{p}_{\text{cam,start}}$.
(2) Capture image and send it to the vision system.
(3) Vision system detects object in the (2D) image, and returns this position to the AIMM.
(4) **IF** *the AIMM does not receive a valid position*
THEN
- Move camera one step along the line from $\boldsymbol{p}_{\text{cam,start}}$ to $\boldsymbol{p}_{\text{cam,end}}$.
- **IF** *the camera already was at* $\boldsymbol{p}_{cam,end}$
THEN the skill has failed
ELSE continue at 2.

(5) Calculate a 3D line from the camera through the (undistorted) image location, received from the vision system.
(6) Calculate the intersection point between the line and the horizontal plane with the (taught) height h_{detect}. This gives the 3D position of the object in camera space.
(7) Transform the object position from the camera space to the robot's base space.
(8) Replace the height of the position with the (taught) value, h_{grasp}, to make the robot grasp the object at a suitable location.
(9) Project the object position on the line between $\boldsymbol{p}_{\text{cam,start}}$ and $\boldsymbol{p}_{\text{cam,end}}$.
(10) **IF** *the projected object position is between* $\boldsymbol{p}_{cam,start}$ *and* $\boldsymbol{p}_{cam,end}$
THEN pick up the object at the calculated 3D position
ELSE the object position is not in the acceptance region, continue at 2.

Table 1. Algorithm for execution of vision based pick skill.



(a) Search for rotor caps (b) Pick up of rotor cap

(c) Search for rotor caps (d) Pick up of rotor cap

Fig. 3. Execution of vision pick skill. The robot captures images while moving in (a). The images are send to the quality control system shown in Figure 1, and whenever a rotor cap is detected, the robot picks it up, as shown in (b). Figures (c) and (d) show the same skill executed at a different location.

is mainly done at the vision system, and the robot itself only needs to know the name of the particular test to perform. In the skill framework, quality control can in general be viewed as a post-condition check, and if this fails, appropriate handling must be implemented. For the tests described here, this can either be to report an error, or to wait a short while and try again. The algorithm for execution of the developed quality control skill is shown in Table 2.

(1) The AIMM signals to vision system to perform (taught) quality control.
(2) Vision system performs control, and replies success/failure.
(3) **IF** *success*
THEN the AIMM continues
ELSE perform appropriate error handling (wait and try again, or call operator).

Table 2. Algorithm for execution of quality control skill.

2.5 Fast calibration

As mentioned in the introduction, the purpose of this skill is to provide calibration in three dimensions, faster than the existing calibration approaches developed by Hvilshøj et al. (2010), which have durations of 10 seconds and above. This is attempted by using the Kinect-like camera Asus Xtion Pro Live, that provides calibrated and undistorted images in both RGB and depth. In our approach, the calibration is implemented as a unique skill, thus having both a teaching and an execution phase. The phases are, however, almost identical. The purpose of both teaching and execution is to find the coordinate system of the (fixed) QR code; the QR frame. Subsequently all locations must be given relative to this frame.

When initiating the calibration skill, the RGB images from the Xtion camera are searched for QR codes. There are several libraries available that provide this functionality, and here *zbar* is chosen, because this directly provide the location of the corners in the images. When a QR code has been found, the depth at each corner of the QR code is averaged over a number of images, and the QR code's coordinate system can then be calculated as:

$$\boldsymbol{x} = \frac{\boldsymbol{c}_0 - \boldsymbol{c}_3}{|\boldsymbol{c}_0 - \boldsymbol{c}_3|} \tag{1}$$

$$\boldsymbol{y} = \frac{\boldsymbol{c}_2 - \boldsymbol{c}_3}{|\boldsymbol{c}_2 - \boldsymbol{c}_3|} \tag{2}$$

$$\boldsymbol{z} = \frac{\boldsymbol{x} \times \boldsymbol{y}}{|\boldsymbol{x} \times \boldsymbol{y}|} \tag{3}$$

where $\boldsymbol{c}_{\text{n}}$ is the location of the n'th corner of the QR code, and the corners are numbered clockwise.

To be able to work in this coordinate system it must be converted into a complete transformation matrix. This is done by calculating a *translation* and a *rotation*. The translation $\boldsymbol{t}_{\text{QR}}$ is defined by the center of the QR code, and is thus calculated as the mean of the corners:

$$\boldsymbol{t}_{\text{QR}} = \sum_{n=0}^{3} \frac{\boldsymbol{c}_{\text{n}}}{4} \tag{4}$$

The rotation $\boldsymbol{r}_{\mathrm{QR}}$ is best defined as Euler angles, which can be calculated directly from the axes. The transformation from the QR code to the camera, ${}^{\mathrm{C}}_{\mathrm{QR}}T$, can be determined by combining the translation and rotation. The desired transformation is between the robot's base and the QR code, ${}^{\mathrm{B}}_{\mathrm{QR}}T$, and this is computed as:

$$ {}^{\mathrm{B}}_{\mathrm{QR}}T = {}^{\mathrm{B}}_{\mathrm{C}}T \cdot {}^{\mathrm{C}}_{\mathrm{QR}}T \tag{5} $$

where ${}^{\mathrm{B}}_{\mathrm{C}}T$ is the (fixed) transformation from the camera to the robot's base.

Finally, the coordinate system given by this transformation is applied to the robot. The exact sequence of execution of the calibration skill is given in Table 3.

(1) **WHILE** *correct QR code not found*
- Search for QR code in RGB images
- Read QR code
- **IF** *the text of the QR code matches taught string* **THEN** exit while loop

(2) Capture a number of depth images.
(3) **FOR** *each corner of the QR code*
- At the location of the corner, calculate the mean of the depth values (ignore 0-values).

(4) **IF** *one or more corners have no depth values* **THEN** the skill has failed. Exit.
(5) Define a coordinate system at the QR code as in Equations (1)-(3).
(6) Calculate the **translation** of the QR code $\boldsymbol{t}_{\mathrm{QR}}$ as the mean of the corners.
(7) Calculate the **rotation** of the QR code's coordinate system $\boldsymbol{r}_{\mathrm{QR}}$ in Euler angles.
(8) Combine translation and rotation into a transformation matrix, ${}^{\mathrm{C}}_{\mathrm{QR}}T$.
(9) Calculate the transformation from the QR code's coordinate system to the robots base coordinate system, ${}^{\mathrm{B}}_{\mathrm{QR}}T$, as in Equation (5).
(10) Set the robots frame base to ${}^{\mathrm{B}}_{\mathrm{QR}}T$.

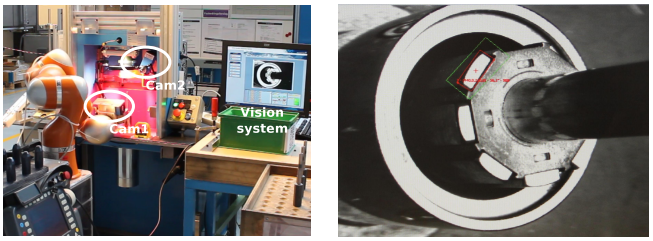
Table 3. Algorithm for execution of calibration skill based on QR codes. The Asus Xtion Pro Live was used for capturing RGB and depth images.

3. RESULTS

The three applications of computer vision have all been implemented on Aalborg University's AIMM *Little Helper*, and tested in a real-life industrial environment at a Grundfos factory. The vision pick skill was able to successfully pick an arbitrary number of rotor caps from two different locations, as shown in Figure 3. The precision was within ±5 mm, which was sufficient to correctly place the rotor caps at the desired locations afterwards.

The quality control was used for a variety of different tests. The application of this integration is only limited by the capabilities of the vision system itself, which is not described here. An example is shown in Figure 4, where it is detected that a magnet has been correctly placed beside the rotor core.

The setup for using the calibration skill is shown in Figure 5. The switch in the Figure is used to enable and disable the conveyor belt. The purpose of the calibration is here to make it possible for the robot to operate the switch,



(a) Assembly (b) Quality control

Fig. 4. Quality control setup. The robot is performing assembly tasks to the left in (a), while the quality control system is running externally, shown on the screen to the right. Figure (b) shows a close up of the result. The green box is the region of interest (ROI), and the red marking is the detected magnet.

and the position of the switch can thus be considered as a position of interest. The position of the QR code relative to the position of interest of course affects the calibration precision, and especially three factors affect the overall precision:

(1) The position estimate of the corners of the QR code the the cameras RGB image. These positions can be determined with sub-pixel accuracy, and at a distance of about 1 m as used in this setup, the precision of the corners is within ±1 mm.
(2) The relative error in the depth values at the corners for repeated measurements. The depth sensor in the Asus Xtion is the same as in the Kinect, and the absolute precision of the depth values provided by the Kinect has been shown to be within ±10 mm for distances between 0.8 m and 3.5 m when used indoor (El-laithy et al., 2012). No data are available on the relative repeatability error, but it has proven to be significantly smaller.
(3) The relative error in the depth values between the corners. No data are available on this precision, but this has also proven to be significantly less that the absolute error.

Especially the third factor; the relative error between the corners, is of interest, because this will cause the coordinate system at the QR code to have a slightly wrong orientation. A wrong orientation makes the error increase the longer the distance between the QR code and

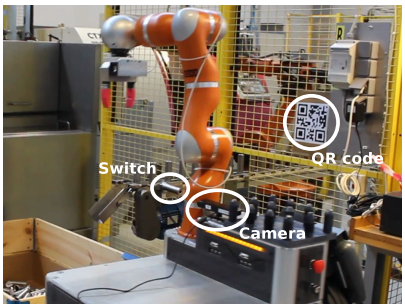


Fig. 5. Fast calibration using the Asus Xtion camera featuring calibrated RGB and depth images. The camera detects the pose of the fixed QR code, and subsequent movements with the robot are corrected accordingly.

the position of interest, and this is also what was found to be the case in the test scenario. Although no formal measurement of the precision has been carried out, visual inspection has shown that the precision is at least ±10 mm at any position. This precision proved to be sufficient to make it possible to operate the switch. In Table 4, the proposed calibration is compared to existing methods.

4. DISCUSSION

Integration of computer vision abilities into a skill based framework proved to be possible, and in this paper, three applications were successfully implemented. In particular the implemented quality control is very generic, and using the developed TCP/IP based protocol, the vision system could be changed without making any changes to the robot itself. This is also the case for the pick skill; however this has in the current implementation some limitations. It is currently assumed that the items to pick up are approximately placed in a line, as is for instance the case on a conveyor belt. Thus during teaching, the start and end location of the camera are taught. A further development should make it possible to define an arbitrary search region during teaching of the skill. For this, an optimal search pattern should automatically be calculated by the robot, taking into account that objects closest to the robot must be picked first. Positions and orientations of the camera during search should also be automatically determined.

The implemented calibration skill makes it possible to perform a very fast calibration compared to existing methods. This is especially important for industrial robots that are moving frequently between workstations. The precision was sufficient to perform the experiments described here, but for high-precision tasks it will be insufficient. There are two obvious ways of doing this:

- The Asus Xtion camera used, does in principle support RGB images with a 1280×960. A bug in the available open source drivers limited, however, the available resolution in our implementation to 640×480. Use of the full resolution images will definitely increase the precision of the QR code detection.
- From the depth image, only the four corner points were used. A better performance could be achieved by using the entire surface of the QR code, for instance by applying the RANSAC algorithm to filter out outliers.

It is impossible to say how much the precision can be improved. However, an experiment should be carried out to determine the precision exactly.

Method	**Duration**	**Precision**
Haptic[1]	30-45 sec	±1.0 mm
High speed[2]	10 sec	±1.0 mm
High precision[2]	60 sec	±0.1 mm
Proposed method	<1 sec	< ±10 mm

Table 4. Comparison of calibration methods. 1 are from Pedersen (2011); 2 are from Hvilshøj et al. (2010).

ACKNOWLEDGEMENTS

This research was partially funded by the European Union project TAPAS under the Seventh Framework Programme. The applications developed here was tested in an industrial environment at a factory owned by the Danish company Grundfos, and this was only possible due to a very close collaboration. Especially, thanks to J. Bigum and R. Larsen for assisting in developing the applied computer vision system.

REFERENCES

Alici, Gürsel and Shirinzadeh, Bijan. A systematic technique to estimate positioning errors for robot accuracy improvement using laser interferometry based sensing. *Mechanism and Machine Theory*, 40(8):879–906, 2005.

Benavidez, Patrick and Jamshidi, M. Mobile robot navigation and target tracking system. In *System of Systems Engineering (SoSE), 2011 6th International Conference on*, pages 299–304. IEEE, 2011.

Bjorkelund, Anders; Edstrom, Lisett; Haage, Mathias; Malec, Jacek; Nilsson, Klas; Nugues, Pierre; Robertz, Sven Gestegard; Storkle, Denis; Blomdell, Anders; Johansson, Rolf; Linderoth, Magnus; Nilsson, Anders; Robertsson, Anders; Stolt, Andreas, and Bruyninckx, Herman. On the integration of skilled robot motions for productivity in manufacturing. In *Assembly and Manufacturing, IEEE International Symposium on*, pages 1–9, 2011.

Björkelund, Anders; Malec, Jacek; Nilsson, Klas, and Nugues, Pierre. Knowledge and skill representations for robotized production. In *Proc. 18th IFAC World Congress*, 2011.

Brooks, Rodney. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14–23, March 1986.

El-laithy, RA; Huang, Jidong, and Yeh, Michael. Study on the use of microsoft kinect for robotics applications. In *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pages 1280–1288. IEEE, 2012.

EUROP. Robotics visions to 2020 and beyond - the strategic research agenda for robotics in europe (SRA). Technical report, EUROP, 2009a.

EUROP. Application requirements. Appendix to: Robotics visions to 2020 and beyond - the strategic research agenda for robotics in europe (SRA). Technical report, EUROP, 2009b.

Gat, Erann. On three-layer architectures. In Kortenkamp, David; Bonnasso, Peter R., and Murphy, Robin, editors, *Artificial Intelligence and Mobile Robots*, pages 195–210, 1998.

Guizzo, Erico and Ackerman, Evan. The rise of the robot worker. *Spectrum, IEEE*, 49(10):34 –41, october 2012. ISSN 0018-9235. doi: 10.1109/MSPEC.2012.6309254.

Hvilshøj, Mads; Bøgh, Simon; Madsen, Ole, and Kristiansen, Morten. Calibration techniques for industrial mobile manipulators: Theoretical configurations and best practices. In *Robotics (ISR), 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pages 1–7. VDE, 2010.

Hvilshøj, Mads; Bøgh, Simon; Madsen, Ole, and Kristiansen, Morten. The mobile robot "Little Helper":

Concepts, ideas and working principles. In *ETFA*, pages 1–4, 2009.

Klank, Ulrich; Pangercic, Dejan; Rusu, Radu Bogdan, and Beetz, Michael. Real-time CAD model matching for mobile manipulation and grasping. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 290–296. IEEE, 2009.

León, Adrián; Morales, Eduardo; Altamirano, Leopoldo, and Ruiz, Jaime. Teaching a robot to perform task through imitation and on-line feedback. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 549–556, 2011.

Murray, Don and Little, James J. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8(2):161–171, 2000.

Nava, Federico; Sciuto, Donatella; Santambrogio, Marco Domenico; Herbrechtsmeier, Stefan; Porrmann, Mario; Witkowski, Ulf, and Rueckert, Ulrich. Applying dynamic reconfiguration in the mobile robotics domain: A case study on computer vision algorithms. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 4(3):29, 2011.

Nilsson, Nils J. *Principles of Artificial Intelligence*. Morgan Kaufmann Publishers, January 1993. ISBN 0934613109.

Pedersen, M. Rath. Integration of the kuka light weight robot in a mobile manipulator. Master's thesis, Aalborg University, 2011.

Shotton, Jamie; Fitzgibbon, Andrew; Cook, Mat; Sharp, Toby; Finocchio, Mark; Moore, Richard; Kipman, Alex, and Blake, Andrew. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304. IEEE, 2011.

Siegwart, Roland and Nourbakhsh, Illah R. *Introduction to autonomous mobile robots*. MIT press, 2004.

Tölgyessy, Michal and Hubinskỳ, Peter. The kinect sensor in robotics education. In *Proceedings of 2nd International Conference on Robotics in Education*, 2010.

Zhang, Zhengyou. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

Human Assisted Instructing of Autonomous Industrial Mobile Manipulator and its Qualitative Assessment

C. Schou, C. F. Carøe, M. Hvilshøj, J. S. Damgaard, S. Bøgh, O. Madsen

Department of Mechanical and Manufacturing Engineering, Aalborg University
Fibigerstræde 16, 9220 Aalborg Ø, Denmark
e-mail: {cs, jsd, sb, om}@m-tech.aau.dk

***Abstract*—Based on current research into skill-based robot programming and human-robot interaction this paper presents a practical implementation and testing in a real life industrial environment. An Autonomous Industrial Mobile Manipulator (AIMM) offers a high level of hardware flexibility compared to stationary robot cells. In order to benefit from this flexibility in the industry the demand for new approaches to operating and programming new tasks is inevitable as traditionally robot programming is time-consuming and often requires robot expertise. Research within this topic has proposed a skill-based approach, where robot programming is generalized into a selection of skills. By the introduction of task-level-programming, robot programming is no longer based on motions of the robot, but based on object-related manipulations. In this paper the task-level-programming approach along with a three layered architecture is the foundation for the underlying work. This paper presents a human assisted intuitive programming interface based primarily on physical human interaction with the robot. This programming interface is a step towards bringing the robot programming and instructing from the robotics experts to the production floor operator. The system has been tested in two different scenarios, one at Aalborg University and one at Grundfos A/S. In both scenarios shop floor operators with limited robotics knowledge have programmed an industrial pick and place task. After receiving a 15 minutes introduction both operators successfully programmed a pick and place task in less than 5 minutes.**

I. Introduction

During the past decade the production paradigm has started changing from mass production and Dedicated Manufacturing Lines towards mass customization and more flexible and agile production systems. This change is necessary in order to cope with the globalization of markets, the trade instability, and the explosion of product variety, which are stressing the time to market and the need for increased adjustment and responsiveness [1]. From this shift a demand for more flexible production equipment emerges, not least in the field of robotics. A solution to the inflexibility of traditional robots is to mount the robot to a moveable platform and thus creating an Autonomous Industrial Mobile Manipulator (AIMM). An AIMM presents a highly flexible and automated production resource, but this technology has not made it to the industry yet as it still faces several issues. One of these issues is how the AIMM is instructed (programmed) to a new task. Given the flexible concept of AIMM this instruction must be available to the shop floor personal [2], [3], [4], hence it must not require extensive robotics or programming expertise. Furthermore the instructing should be relatively quick and robust.

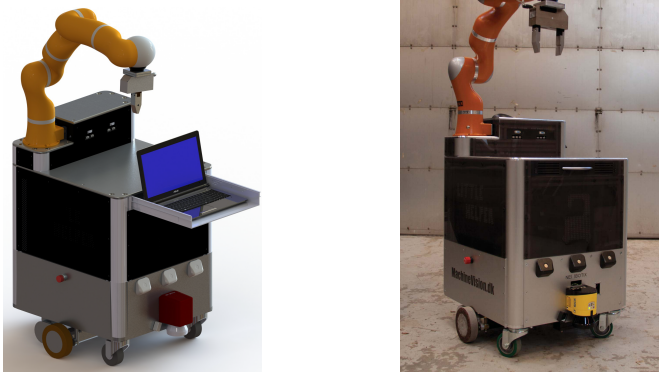
To meet these demands traditional robot programming is not sufficient. Instead the programming must be brought from a robot-specific programming language to a higher level of abstraction. A paradigm that have proven feasible in research is the task-level-programming, as it is focused on object related goals and not the robot motions to achieve it.
Several researchers have attended the issue of instructing mobile manipulators with the approach of automatic sequencing of actions. One of the first approaches was the STRIPS-rules presented by R. Fikes and N. Nilsson in 1971 [5] in which a problem space is formulated by an initial state, a goal condition and a set of actions. From the problem space a search tree is formulated and the ”optimal” route (or sequence) from the initial state to the goal is identified.
Another approach is the Knowledge Integration Framework (KIF) [6], [7], [8]. In this work the sequencing of a task relies on knowledge from previous and similar tasks stored in the knowledge framework KIF. The knowledge obtained from one task is stored as a ”recipe” and is used to generate sequences for similar tasks.
Task-level-programming is also being used without the purpose of automatic task generation, but to obtain a higher level of abstraction in programming methods for robotics. This is the focus of the SKill Oriented Robot Programming (SKORP) [9]. Research at Aalborg University has focused on the identification, formulation and integration of skills aimed at tasks in the industrial manufacturing environment [10], [11], [12].

Inspired by the SKORP and building on the current research at Aalborg University this paper presents an intuitive interface for instructing AIMMs, without requiring extensive robotics training. This interface is based on a graphical Man-Machine-Interface (MMI) intended for a portable tablet and direct interaction with the manipulator. In order to create a higher level of abstraction in relation to traditional robot programming, the task-level-programming is used. The purpose of this interface is to demonstrate a functional robot programming method that can be used by the shop floor operators in the production environment.

This paper is organized as follows: Section II introduces "Little Helper", an AIMM research platform from Aalborg University. In section III the concept of task-level-programming is presented. The intuitive instruction interface is described in section IV and section V describes two tests, where shop floor operators use the intuitive interface to instruct Little Helper to perform industrial pick and place tasks. The results are presented in section VI.

II. Little Helper

In 2007 the first AIMM at Aalborg University was designed and assembled. The fundamental vision was to create a "little helper" to assist the shop floor operators in industrial manufacturing by attending some of the simple repetitive tasks in the production. From this vision the mobile robot and the project itself got the name "Little Helper" [3]. Today a total of three versions of Little Helper has been built at Aalborg University, but only two of them remain in use. The latest version, Little Helper 3 from 2012, is shown in figure 1.



(a) CAD drawing. (b) Photography.

Fig. 1: Little Helper 3, an AIMM at Aalborg University designed for attending complex assembly tasks.

Little Helper 3 is primarily assembled from commercial-of-the-self (COTS) components, but a custom aluminium frame structure to interface the different components had to be designed. The key components are:

- Neobotix MP-655 non-holonomic platform
- KUKA Light Weight Robot
- Schunk WSG-50 electric parallel gripper
- Two onboard computers, one of them an accessible laptop
- Wireless emergency stop

The software and thus control of Little Helper uses a distributed architecture where the low-level and real-time control of each device is carried out on the device itself. The Robot Operating System (ROS) [13] is used to link the distributed nodes. A central node utilises all the distributed device nodes and thus creates the higher level of control. This node also incorporates the user interface including task-level-programming, which is described in the next section.

III. Task-level-programming

If the AIMM is to be programmed by an operator with limited robotics knowledge at the shop floor during production runtime, the programming interface must be easier and faster to use, than conventional robot programming interfaces. In order to do this it is chosen to bring the robot programming from the level of simple device specific commands to a task focused level. The approach of task-level-programming is divided into three layers consisting of device primitives, skills, and tasks; inspired by [14]. Figure 2 shows the setting and the interaction between the individual layers. Tasks, skills, and device primitives are described in the following sections.

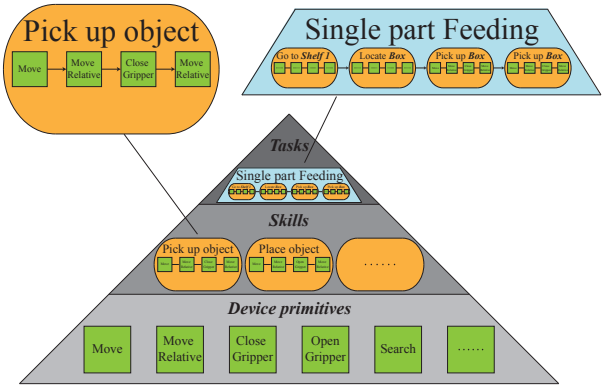


Fig. 2: This figure shows the three layers of abstraction used on Little Helper. The lowest level is the device primitive layer, which is basic functionalities and motions of the different devices; hence this layer is the closest to real-time. The middle layer is the skill layer, which are object-centred capabilities of the AIMM as a whole. Sequencing and parameterizing skills creates a task and thus the layer above skills. This architecture is inspired by [14].

A. Device Primitives

Device primitives are described as basic motions and functionalities e.g. *Grasp*, *Move to XYZ* or *Search in x-direction*. The motions or functionalities of a skill is conducted as a result of the device primitives. A device primitive is declared as a command conducting a motion or functionality while the level below is described as the hardware embedded driver.

B. Skills

One of the key elements in bringing robot instruction from a robot-specific programming language to a higher abstraction level is skills. On one hand the skills represent the foundation of the task, and thus the building blocks used by the operator. On the other hand skills are defined as a higher level abstraction of functionalities and motions of the individual devices of the robot; that is a higher level abstraction of device

primitives. A skill utilises these device primitives as motions and combining them with sensor input, advanced mathematics and advanced robotics the skill manipulates an object in order to achieve a production-related goal. That is, a skill is an object-oriented capability of the AIMM as a whole, i.e. ”pick”, ”place”, ”rotate” etc.
It is in the skill layer the robotic experts can bring advanced robotics software into the system, yet keeping an object-oriented, and thus task-oriented, level of abstraction for the operator, compared to a level of simple device specific commands.
Based on the skill definition a general skill model is established in figure 3.

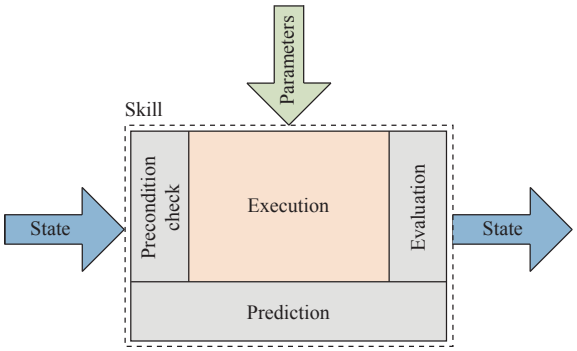


Fig. 3: Skill model. A skill is composed of an execution routine, preceded by a precondition check to verify that the initial state is feasible to this skill. Subsequent to the execution routine a postcondition check verifies the outcome of the skill and compares it to a predicted outcome. [10]

A skill relies on a defined motion sequence, but the motions themselves are adapted to a specific task by a set of parameters, thus making the skills generic within a certain scope. The parameters are established during the instructing of the robot and stored in a task file along with the skill sequence.
During execution of the skill these parameters are given to the system together with a set of state variables. A precondition check is conducted by examination of the given input parameters compared to the measured state. If the precondition check is performed successfully the execution part is conducted. In other words, the precondition check serves as a safety net. The postcondition check is conducted based on a prediction of the desired goal and an evaluation of the outcome. If the comparison between the evaluation and the predicted outcome is consistent within a given range, the postcondition check is successful. As outcome the state variables are changed and updated based on the accomplished skill.

C. Tasks

A task is described as a sequence of skills and contains an overall goal e.g. *pick up the rotor at station A*. In this way a task is established on the basis of a library of skills. A task is defined as a sequence of skills each parameterised to the specific task and thus the task itself is merely a file containing the sequence of skills and the parameters for each skill. The task has a set of measurable state variables that are changed continuously during the skill sequence.

IV. User Interface

The user interface on Little Helper is structured around the concept of task-level-programming and skills. It is apparent that the purpose of instructing a new task is to later execute it and this implies that a routine for obtaining these parameters must be defined. This is met by implementing both a programming routine and an execution routine in each skill. By giving each skill a dedicated programming routine it is ensured that the needed parameters for the execution routine are obtained. This coherency in the skills is also reflected in the user interface, which also includes a section for programming and a section for execution. The correlation between task programming and execution is illustrated in figure 4

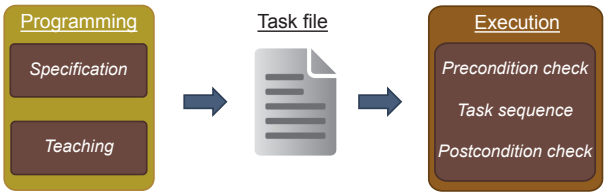


Fig. 4: Illustration of the correlation between the programming part and the execution part of a task. During the programming part the sequence of skills is established and a set of parameters is obtained through a programming routine. This information is stored in a task file, from which the task can later be executed. The programming part is divided into a specification phase and a teaching phase.

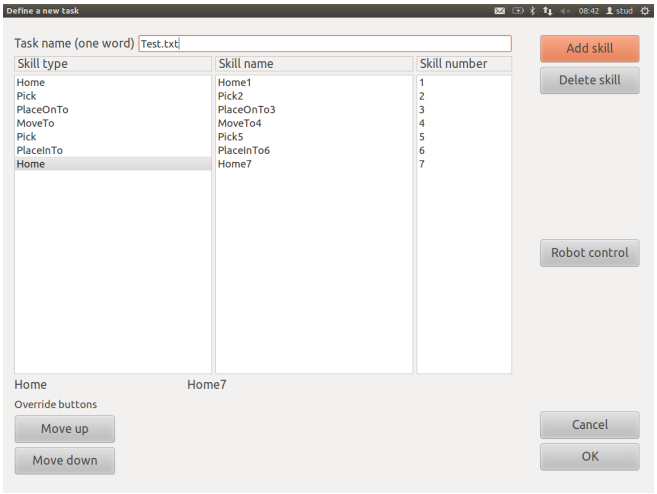
A. Programming

As shown on figure 4 the programming part is divided into a specification phase and a teaching phase. During the specification phase the sequence of skills is chosen and partly parameterised, and during the subsequent teaching phase the locational parameters are obtained through direct interaction with the AIMM.

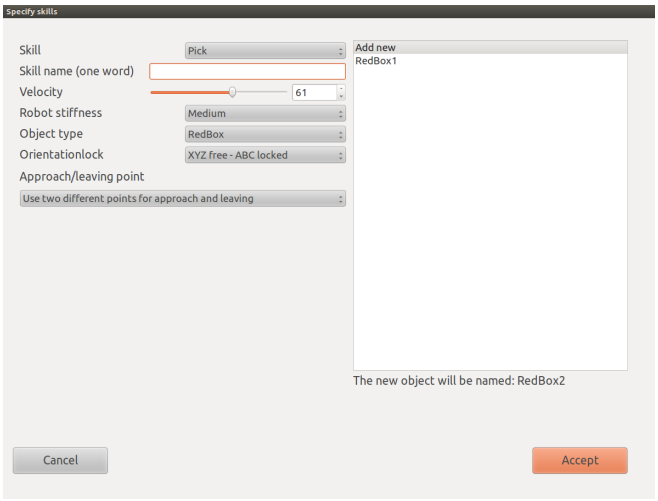
Specification
The specification phase is conducted in a graphical MMI on a PC or on a tablet thorough a remote desktop application. During the specification phase the skill sequence is chosen from the library of skills and several parameters are obtained through user input. These inputs are given by clicking and selecting in boxes and drop-down menus and to simplify the interface several parameters are preselected to standardised values, which are concealed in an advanced tab.
Figure 5 shows two screenshots of the graphical MMI; a screenshot of an established sequence and a screenshot of the parameters selected for a skill.

During the specification phase several parameters are given for each skill. Some of these parameters correlate to the execution of the skill where others correlate to how the following teaching phase is conducted.

Teaching
Once the skill sequence has been configured the teaching phase is commenced, during which the operator directly



(a) Skill sequence



(b) Skill parameters

Fig. 5: Two screenshots of the graphical MMI. The upper figure shows the window for creating a new task where a skill sequence has been established. The lower figure shows the window for selecting a skill along with parameters for it.

interacts with the manipulator of the AIMM. During the teaching phase each skill in the skill sequence is taught sequentially. In this way the teaching sequence correspond directly to the execution sequence and thus creates a clear overview of the progress and outcome. The teaching of the skills follows the teaching routines for each skill, during which the operator is guided through written instructions in the graphical MMI and ”beep” sound outputs. The KUKA LWR has a build-in control mode called ”Gravity Compensation”. In this mode the manipulator only compensates for its own weight (including tool) by use of integrated torque sensors in each joint and thus any measured external force results in an acceleration. In Gravity Compensation mode the operator can directly interact with the manipulator and pilote it to a given coordinate. Furthermore these force-sensors facilitate the measuring force applied to the end-effector, which provide a convenient method of obtaining user input.

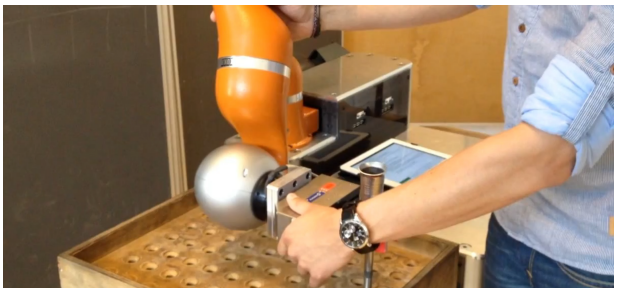


Fig. 6: Photograf of an operator interacting with the manipulator and piloting it to a coordinate.

Teaching a 3D coordinate is central in many skills. This starts by the operator applying force to the end-effector to start the teaching process. This puts the manipulator into gravity compensation mode and the operator drags the tool to the location. The location is stored when the tool center point (TCP) is held steady for three consecutive seconds.

B. *Execution*

When implementing AIMMs on a larger scale in an industrial production environment, determining which tasks it should carry out during the day should be handled by a centralised mission planner with access to the Enterprise Resource Planning (ERP) system. In this way the AIMM is automatically assigned to the tasks, where the capacity needs are the highest and thus the full benefit of the flexibility of the AIMM is achieved.
In several scenarios the operator might need to execute a task or assign a small mission to the AIMM, i.e. to test a newly programmed task or if the AIMM is used to assist the operator for shorter periods of time.
The graphical MMI incorporates both functionalities. It has an automatic mode, where it responds to missions from a mission planner, and it has a simple execution mode, where the operator selects one or several tasks to execute.

In both cases the system will open the selected task file, interpret it and execute the given skills with the parameters from the task file, while following the skill model in figure 3.

V. Test of the User Interface

The presented user interface has been tested in a scenario at Aalborg University and in a scenario at the Danish pump manufacture Grundfos A/S in Bjerringbro, Denmark. In both scenarios an operator is given a 15 minutes introduction, including both an introduction to the hardware devices of Little Helper, the different skills, the interaction with the manipulator, the graphical MMI and the task. In the actual programming phase the operators works independently, but a robotics expert stands by to provide help when requested by the operator.
The operators have both a laptop with the graphical MMI at their disposal along with an iPad tablet making the MMI accessible through a remote desktop application.

A task is successfully taught if the execution of the task achieves the specified goal. It is observed how long time the programming takes and how many times the operator requests help from the expert.

A. Aalborg University scenario

In the scenario at Aalborg University a shop floor operator from the machine shop at the Department of Mechanical and Manufacturing engineering, has been challenged to program a pick and place task. The task is a replication of an actual industrial production scenario at Grundfos where a rotor-cap for a Grundfos SQFlex pump arrives on a conveyor. Little Helper will pick this rotor-cap and place it in a fixture on itself. Figure 7 shows the setup of the task.

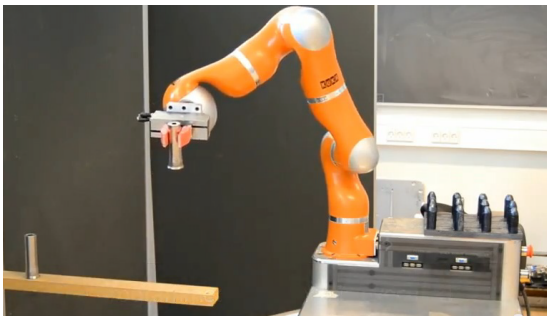


Fig. 7: Setup of the pick and place task in the Aalborg University scenario.

The task can be described as follows:

1) Align the gripper to the rotor cap
2) Pick the rotor cap from the conveyor
3) Align the gripper and the picked object to the fixture
4) Place the rotor cap on the fixture

B. Grundfos scenario

The Grundfos scenario takes place in the production facility of the Grundfos SQFlex submersible pump. The shop floor operator is from Grundfos. To avoid interference with the running production it is chosen to use the components from the Cranfield benchmark to perform a pick and place task. The setup at Grundfos and the Cranfield components are shown in figure 8. In the test a square component from the Cranfield test is picked from one of the two main plates and placed in the other main plate.

The task can be described as follows:

1) Pick the square from the first main plate
2) Place the square into the other main plate

VI. Results

In this section the results of both test scenarios are presented.

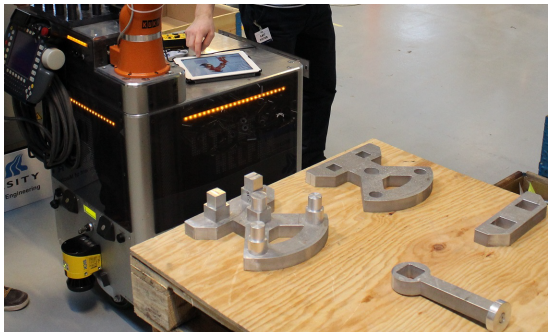


Fig. 8: Setup of the pick and place task in the Grundfos scenario. The Cranfield components are shown in the lower right corner. The two main plates are placed next to each other, while the five square and cylindrical components are placed in one of the main plates.

A. Aalborg University Scenario

In the scenario from Aalborg University the operator chose the following skill sequence to complete the task of picking a rotor-cap from a conveyor belt and placing it in a fixture on the platform of the Little Helper.

1) Home
2) MoveTo
3) Pick
4) MoveTo
5) Place
6) MoveTo
7) Home

A fast-forward video of the operator instructing the task is found via the link provided in the footnote [1]. During the test the operator successfully programmed the task in the first attempt using 4 minutes and 13 seconds. From the following execution the programmed sequence is verified and although it is less refined than the work of a robotics expert, the task is executed with success. During the programming of the task the operator requested help once.
This paper will not elaborate on the various skills used in the scenarios, but it should be noted, that the ”MoveTo” skill is used to both align the gripper to the object and to ensure collision free cartesian paths between locations.
Through the test issues regarding the remote desktop interface used to transfer the MMI interface to the tablet were experienced and consequently the intuitive interface was effected by this. As a result the specification phase was conducted on the laptop while the iPad was used during the teaching part where the issue is counterbalanced by the portable benefits of the iPad.
In conclusion of the test, the operator is quoted:

”Using this system for a few hours, I reckon that I would master it sufficiently to configure a more complex task.”

[1] A fast-forward video of the operator instructing the task: http://youtu.be/hmFPnNIifZM

"It wouldn't take long before I don't need the instructions on the iPad in order to interact with the manipulator."

B. Grundfos Scenario

The operator in the Grundfos scenario selected the following skill sequence:

1) Home
2) Pick
3) MoveTo
4) MoveTo
5) PlaceInto
6) MoveTo
7) Home

The operator successfully programmed the task in the first attempt using 4 minutes and 48 seconds. As seen in the skill sequence he used two "MoveTo" skills consecutively, as he during the specification phase couldn't recall the motion sequence of the "Pick" and "PlaceInto" skills.
It was observed, that at the production scene the noise-level was relatively high and as a result the sounds intended to inform the operator during the teaching phase could not be heard. This significantly increased the complexity during the teaching phase. As a consequence the operator was not completely confident in the teaching of the different skills, even though the sequence was actually executed successfully later.
The operator did not require any assistance during the programming of the task, but he was unable to execute the programmed task due to an improperly chosen name for the task file. This is merely a technical issue with the software.

After the test the Grundfos operator gave several comments on his experience. First of he found it rather easy to use and not difficult to learn. Despite that he encountered a few difficulties during the teaching phase, he was certain that with a few hours of practice he could program pick and place operations easily. Concludingly he had several comments on how to improve the interface especially during the teaching phase. He suggested using visual instructions in the form of video or pictures instead of the written instructions during the teaching phase. This would significantly clarify what he (as an operator) was to do next. Furthermore he suggested that the operator is presented with the a visual overview of the entire motion sequence of a skill prior to teaching it.

VII. Conclusion and future work

In this paper a user interface to program and execute industrial tasks on the Little Helper, an AIMM from Aalborg University, has been presented. The Foundation for the interface is a three layered architecture with skills as the building blocks for the operator to compose a task from. The user interface is aimed at being an intuitive interface making the programming of industrial tasks available to a shop floor operator. Even though the implementation of a new task is configured in a user-friendly graphical MMI it should be noticed that the teaching of a skill still requires general knowledge about robotics. In this way the operator should be able to predict singularities, speed limits, torque limits, collisions, and joint limits.
Dividing the programming part into separate phases, hence the specification phase and a teaching phase, is disadvantageous for a complicated task as it is difficult to predict the individual skills in the total sequence when selecting the entire sequence in the specification phase.
In both tests the shop floor operators with very limited robotics knowledge were able to program a pick and place task in the first attempt in under 5 minutes given only a 15 minutes introduction. Even though a few issues was encountered, both the operators were able to program a task that could be executed to achieve the desired goal.
Both operators found the interface intuitive and easy to use and both felt confident that with a little practise they would no longer need the instructions.

The conclusion is, that with the presented system the programming of industrial pick and place tasks are both expedite and at the same time brought to a level where robotics expertise is no longer needed. This makes the technology available to the shop floor operators. The interface still requires better instructions to support the operator during the teaching phase and basic issues from the robotic domain still needs to be addressed, such as avoiding singularities, avoiding joint limits and choosing collision free paths. As a concept, the presented interface with the task-level-programming and direct human robot interaction presents a platform for making technology available to the shop floor operators. As a result their production insight and knowledge will be transfered to the robot-sequence.
The presented user interface proves, that even though a highly complex and fully automatic skill sequencing system could further ease the robot programming, the idea of letting the shop floor operator instruct the task step-by-step is feasible.

In future work a motion planner will be integrated to automatically generate collision free paths, which will significantly aid the operator in the teaching phase. Furthermore the written instructions during teaching will be replaced with animated figures, videos or voice commands.
Pick and place are very common in industrial productions, but a more substantial and profound test will be carried out on a larger group of operators. Furthermore the tasks will be carried out using traditional robot programming to serve as a reference.

Acknowledgment

The authors would like to thank Grundfos A/S for excellent collaboration and for hosting a demonstration. The authors would also like to thank both operators for participating in this test.

The authors would also like to thank the project TAPAS under the Seventh Framework Programme in the European Union as the research leading up to this paper was partially funded by this project.

REFERENCES

[1] F. Jovane, Y. Koren, and C. Boër, "Present and Future of Flexible Automation: Towards New Paradigms," *Cirp Annals-manufacturing Technology*, vol. 52, no. 2, pp. 543–560, 2003.

[2] EUROP, "The Strategic Research Agenda for Robotics in Europe - Robotics Vision for 2020 and Beyond," 2009.

[3] M. Hvilshøj and S. Bøgh, ""Little Helper" - An Autonomous Industrial Mobile Manipulator Concept," *International Journal of Advanced Robotic Systems*, vol. 8, no. 2, 2011.

[4] "Robotics-enabled Logistics and Assistive Services for the Transformable Factory of the Future (TAPAS)." EU project funded under the European Community's Seventh Framework Programme. Web page: http://www.tapas-project.eu/.

[5] R. E. Fikes and N. J. Nilsson, "Strips: A new approach to the application of theorem proving to problem solving," *Artificial Intelligence*, vol. 2, no. 3–4, pp. 189 – 208, 1971.

[6] A. Björkelund, J. Malec, K. Nilsson, and P. Nugues, "Knowledge and Skill Representations for Robotized Production," in *Proceedings of the 18th IFAC Congress*, IFAC, 2011.

[7] A. Bjorkelund, L. Edstrom, M. Haage, J. Malec, K. Nilsson, P. Nugues, S. Robertz, D. Storkle, A. Blomdell, R. Johansson, M. Linderoth, A. Nilsson, A. Robertsson, A. Stolt, and H. Bruyninckx, "On the integration of skilled robot motions for productivity in manufacturing," in *Assembly and Manufacturing (ISAM), 2011 IEEE International Symposium on*, pp. 1–9, may 2011.

[8] J. Persson, A. Gallois, A. Bjoerkelund, L. Hafdell, M. Haage, J. Malec, K. Nilsson, and P. Nugues, "A knowledge integration framework for robotics," *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pp. 1–8, 2010.

[9] C. Archibald and E. Petriu, "Model for skills-oriented robot programming (SKORP)," pp. 392–402, 1993.

[10] S. Bøgh, O. Nielsen, M. Pedersen, V. Krüger, and O. Madsen, "Does your Robot have Skills?," in *Proceedings of the 43nd ISR (International Symposium of Robotics)*, ISR, 2012.

[11] S. Bøgh, M. Hvilshøj, M. Kristiansen, and O. Madsen, "Identifying and evaluating suitable tasks for autonomous industrial mobile manipulators (AIMM)," *The International Journal of Advanced Manufacturing Technology*, vol. 61, no. 5-8, pp. 713–726, 2012.

[12] M. Hvilshøj, S. Bøgh, O. Nielsen, and O. Madsen, "Multiple part feeding – real-world application for mobile manipulators," *Assembly Automation*, vol. 32, no. 1, pp. 62–71, 2012.

[13] ROS Wiki, "Documentation - ros wiki." URL: http://www.ros.org.

[14] E. Gat, "On three-layer architectures," in *Artificial Intelligence and Mobile Robots*, MIT Press, 1998.

Interactions between Humans and Robots

Evgenios Vlachos
Department of Communication
Aalborg University
Aalborg, Denmark
evlachos@hum.aau.dk

Henrik Schärfe
Department of Communication
Aalborg University
Aalborg, Denmark
scharfe@hum.aau.dk

***Abstract* — Combining multiple scientific disciplines, robotic technology has made significant progress the last decade, and so did the interactions between humans and robots. This article updates the agenda for robotic research by highlighting the factors that affect Human – Robot Interaction (HRI), and explains the relationships and dependencies that exist between them. The four main factors that define the properties of a robot, and therefore the interaction, are distributed in two dimensions: (1) Intelligence (Control - Autonomy), and (2) Perspective (Tool - Medium). Based on these factors, we introduce a generic model for comparing and contrasting robots (CCM), aiming to provide a common platform for robot designers, developers and users. The framework for HRI we propose stems mainly from the vagueness and the lack of clarity that has been observed in the definitions of both Direct and Indirect HRI.**

Keywords— human - robot interaction; robot properties; interactions; operator; autonomy; control; tool; medium

I. INTRODUCTION

The emerging field of Human-Robot Interaction (HRI), which has recently received global scientific attention [1], is a multidisciplinary area that not only encloses the fields of Humanities, Social Sciences, Computer Science and Engineering, but also expands towards directions connected to Education, Medicine and the Life Sciences [2 - 4]. A shift towards a more human-centered design of HRI has been observed, including issues such as perceptions of robots, robot behavior, believability of interaction, and meeting people's expectations [6, 7]. This human-centered perspective does not imply that the constant technical challenges that arise are of trivial importance. On the contrary, they continue to be taken into serious consideration.

Nevertheless, we still have difficulties in unlocking the mechanisms that steer human thought and action, and we still cannot provide a solid well-formulated definition of what a robot is, as the field of robotics is evolving following the Moore's Law exponential curve [5]. Humans and Robots are two entities that our knowledge about them keeps constantly expanding; we are on the process of understanding the former, and exploring the boundaries of the latter. We believe that through the process of interaction we can approach all the above mentioned issues. However, it is important to realize that the nature of HRI is related to, but is different from human – human, or human - computer interaction.

This study proposes a model that allows comparisons and contrasts among robots from all the scientific fields, by exploiting the four main factors that affect the properties of a robot: Control, Autonomy, Tool, and Medium. In the next sections we will justify the reason why we have chosen these four factors, the relationships and dependencies that exist between them, and present a platform for comparing robots with the Compare-Contrast Model (CCM).

II. INTERACTIONS

Interactions between humans and robots that pertain to the flow of information and control can be separated into two main discrete categories (even though the human-robot communication may take several forms) according to their proximity [1, 8].

1) Direct or Proximate Interaction, where humans and robots are co-located (physical interaction).

2) Indirect or Remote Interaction, where humans and robots are dislocated, and are separated spatially, or even temporally (teleoperation / supervisory control / telemanipulation).

With the rise of Human - Centered Robotics, the role of the human started to claim its own space within the area of HRI, and issues like the types of interaction in HRI, which until now looked well established, started to be questioned. A special kind of mediation is required depending on the position and location of the operator of the robot, and the robot itself, in accordance to the surrounding environment. Most of the definitions about Direct Interaction are referring to the communication **between the robot and its surrounding environment** (composed of humans/ other robots/ objects/ nature), while in the Indirect Interaction are referring to the communication **between the robot and its operator** [8]. Until now we have approached the matter of Direct and Indirect Interaction by comparing two different entities. Even in the -so far accepted as- "Indirect" interaction, the communication between the robot and its surrounding environment is still direct. We firmly believe that there should be a distinction between the flow of information, and the flow of control. Fig. 1 depicts schematically the flow of information for these interactions, when on the other hand, the flow of control is strictly limited to the interaction between the operator and the robot, with a direction from the operator towards the robot.

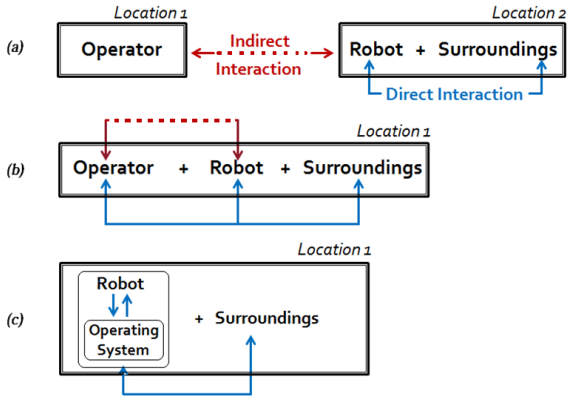


Figure 1. Flow of information in Human-Robot Direct and Indirect Interaction: (a) Operator and Robot Dislocated, (b) Operator and Robot Co-located (operator part of the surroundings), (c) Operator and Robot Co-located (operator part of the robot).

According to the degrees of freedom the operator has, the robot can be more or less controlled, and consequently less or more autonomous. *Autonomy* refers to a robot's ability to accommodate variations in its environment, and is a determining factor of HRI with regards to the tasks a robot can perform, and the level at which the interaction takes place [8]. *Control*, which is the inversely proportional quantity of autonomy, is added to the factors affecting the HRI

Fig. 1 reveals also another factor that affects interactions, the one of location. This is indeed a critical factor, because when the operator and the robot are co-located, the operator is either part of the surroundings, or part of the robot (meaning the operating system). Further research towards that direction may reveal to what extent the presence, or absence of the operator affects interactions between humans and robots. Due to the insufficiency of information on the location factor at this point of research, we decided not to consider it as one of the critical factors for our model.

A. *Interaction Paradigms*

The three primary interaction paradigms are *computer-as-tool* (addressed mainly by the research community of Human-Computer Interaction), *computer-as-partner* (addressed mainly by the research community of Artificial Intelligence), and *computer-as-medium* (addressed mainly by the research community of Computer-Supported Cooperative Work) [9]. The computer-as-tool paradigm extends human capabilities through a tool, the computer-as-partner paradigm embodies anthropomorphic means of communication in the computer, and the computer-as-medium paradigm allows technology to serve as a mediator of communication between geographically distributed environments [10].

In the field of robotics, Artificial Intelligence is spread over almost all its applications, meaning that there are robots serving both as tools [11], and as mediums [12] that can be characterized as partners. Hence, we can consider *Tool* and *Medium* as our next two key factors that affect the HRI. Again, the factors Tool and Medium are inversely proportional, just like a robot built for industrial use, and a robot intended for social interaction. In most of the cases, we will not choose to design our robotic tools with social relational personas [13].

III. CLASSIFICATION OF ROBOTS

The best way to describe the notion of robotics is to look into the different types of robots that exist. Under the prism of Autonomy, Control, Tool, and Medium, we make an attempt to shed some light on the various applications, and fields of practice a robot can be engaged in. We separate these four factors in the dimensions of Intelligence (Autonomy-Control), and Perspective (Tool-Medium).

A. *Intelligence*

The degrees of freedom an operator has, makes the robot more or less *controlled*, and consequently less or more *autonomous*. It is highly likely to find a robot that combines elements from both of these sub-categories [23-29].

1) *Control*

a) Teleoperated, is a remotely controlled robot guided by a human operator who views, and senses the environment through the robot sensors. Such robots are used mainly as mediums for communication (e.g., the Geminoid series [12]).

b) Telepresence, is a robot that provides a two way audio, and video communication for embodied video conferencing using wireless connections (e.g., the Anybots' Virtual Presence Systems [32]).

c) Manually controlled, is a robotic interface contolled in a non-autonomous manner. For example, a hand-operated tool used in surgical operations [33], a gaze-controlled robot [35], a gesture, or voice control robot [36], fall under this sub-category.

d) Brain controlled, is a robot operated through a system that picks up electrical signals stemming from the brain, and translates them into commands [34].

2) *Autonomy*

a) Autonomous, is a robot able to fulfill the given tasks by obtaining information solely from its surrounding environment without human intervention [27]. The human operator is substituted by an operating system located inside the robot. An *Epigenetic*, or a *Developmental Robot* can fall under this category since it uses metaphors from neural development and developmental psychology to develop the mind for autonomous robots [16]. It's a type of robot inspired by the fact that most complex and intelligent biological organisms (as opposed to artificial ones) undergo an extended period of development before reaching their adult form and adult abilities.

b) Semi-autonomous is a robot acting as an autonomous one, except for the occasions that a human operator interrupts its routine, and is involved so as to handle an event, or add perceptual input/ feedback.

c) Neuro controlled, is a robotic system coupled with a network of living neurons coming from the cortex of a vertebrate [37].

B. Perspective

1) Tool, aiming to extend the human capabilities, with Industrial Robots to be the most characteristic example. According to the ISO 8373 definition they are "…automatically controlled, reprogrammable, multipurpose, manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications" [31].

2) Medium, indicating communicative activity mediated via robots. Within this category fall Social Robots which are embodied agents, part of a heterogeneous group (including humans and other robots), and are able to recognize the members of its group, engage in social interaction, communicate within the social and cultural structure, and also learn [17, 28]. Embodiment means establishing a basis for structural coupling by creating the potential for mutual perturbation between system and environment [17]. Social robots are described as relational artifacts that convey intentionality, presenting themselves as having "states of mind" [18, 19]. There are two classes of social robots; the utilitarian robot, and the affective social robot, both assisting humans in achieving better physical, mental and emotional health [19]. Utilitarian robots, or domestic robots, or service robots, are designed to interact with humans mainly for instrumental, or functional purposes, helping them with their tasks. Affective social robots on the other hand, are robots designed to interact with humans on an emotional level, and are used as entertainment, therapeutic companions.

C. Locomotion and Appearance

Two of the features that all the above factors share are Locomotion, and Appearance. *Locomotion* does not constitute a dimension since it has a binary value, either static or mobile, forbidding us to define the degree to which its measurement extends. Consequently, it is not considered as a critical factor, yet we analyze bellow briefly its components.

- *Static Robot*, which usually performs with precision dangerous difficult, or dull repetitive tasks like lifting objects, picking and placing, handling chemicals, or performing assembly work. The term static is interwoven with heavy industrious work, but today exist static robots that perform socially related tasks. One of them is the iCAT platform from Phillips Research [38].
- *Mobile Robot*, which can move and navigate in the real world and can be either autonomous, or controlled. The type of the mobile robot movement varies from floating, swimming, and flying to rolling, crawling, or walking [27].

Maybe the interface is the most important component of a robot because it uncovers immediately the purpose that it serves, and sets the interaction rules. Nevertheless, *appearance* is also not a dimension, and will also not be considered among the critical factors, because its components cannot be valued in one direction. A summary of the available interfaces follows.

- *Mechanoid:* A robot with a machine-like appearance which has no overtly human like features and bears no resemblance to a living creature [20].
- *Zoomorphic:* A robot built to imitate living creatures. For this kind of robots, a zoomorphic embodiment is important for establishing human–creature relationships. Usually their objective is to create robotic "companions" [21].
- *Anthropomorphic (anthrobots):* Anthropomorphism is a term coming from the Greek term "anthropos" for man and "morphe" for form, and is attributing human characteristics to robots aiming to rationalize their actions [30].
- *Humanoid:* A robot which is not realistically human-like in appearance, but possesses some human-like features, which are usually stylized, simplified or cartoon-like versions of the human equivalents, including some or all of the following: a head, facial features, eyes, ears, eyebrows, arms, hands, legs.
- *Android:* A robot which is built to mimic humans both in appearance, and behavior. Androids have a broad range of applications and can sometimes combine the features of various types of robots [20], [22].
- *Caricatured:* The principle of exaggeration is at the heart of caricature. It involves amplifying the distinct features that identify the kinetic display in order to make the content of the behavior more convincing. This involves isolating the features that uniquely identify the content of the expression [14].
- *Virtual:* These robots act like virtual simulators in order to test the software of a robot while the real robot is still at the stage of development. It predicts the result of a command before the command is send to the remote robot [39].

IV. The Compare - Contrast Model

As we have discussed in the previous sections, autonomy and control are two inversely proportional quantities, meaning that the more autonomy a robot has, the less controlled it is. In that case, the *Robot Properties* are depending on the Control and Autonomy Properties the robot encloses. The line tangent to the function *Autonomy = 1/Control* (Fig. 2a) is the hypotenuse of the (always) right triangle that is formed, and represents the Robot Properties. Likewise, tool and medium are also two inversely proportional quantities (*Tool = 1/Medium*) that define the Robot Properties, and a second right triangle is formed (Fig. 2b).

The reasoning process described so far, leads to the following four extreme situations that can characterize a robot: (1) totally Medium - totally Controlled, (2) totally Medium - totally Autonomous, (3) totally Tool - totally Autonomous, and

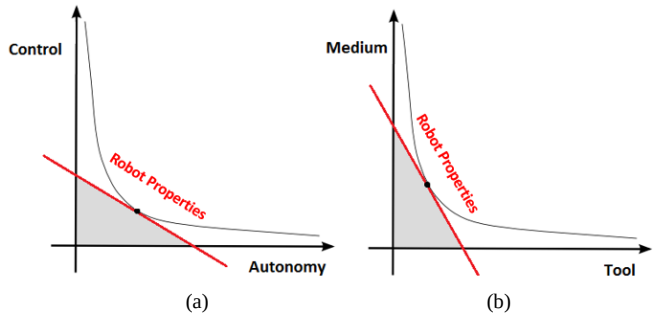


Figure 2. The hypotenuse of each right triangle depicts the Robot Properties for the (a) Intelligence dimension (Control and Autonomy), and the (b) Perspective dimension (Medium and Tool).

(4) totally Tool - totally Controlled. Before we proceed further, we should note that our study is taking into consideration not only the existing robotic technology, but also future scenarios where robots may be a naturally integrated part of human life, or even act independently of humans.

- *To be totally a Medium and totally Controlled.* This is the usual scenario for most of the teleoperated and telepresence robots. The Giraff "caregiving" robot is a characteristic example [15].
- *To be totally a Medium and totally Autonomous.* If a robot is used as a medium, then it cannot take decisions automatically. It is built to communicate messages from one person to another. For instance, an "autonomous" virtual agent with a set of pre-programmed responses, or with the ability of adjusting its behavior to the user via fuzzy logic algorithms, can be described only as a medium and will never obtain total autonomy since it will always be serving its programmer. We can safely state that the Medium factor, and the Autonomous factor are two inversely proportional quantities (Medium = 1/Autonomy).
- *To be totally a Tool and totally Autonomous.* An industrial robot (e.g., a robotic arm) is the perfect example of this scenario.
- *To be totally a Tool and totally Controlled.* When a robot is totally controlled, all and only the intentionality and the capabilities of the operator are transferred to the robot, and mediated to the surrounding environment. On the contrary, a robot as a tool is extending the capabilities of the human, in this case the capabilities of the operator. The robot needs to have at least a very small percentage of automation embedded inside in order to fulfill the expectations of a tool. A robot functioning as a tool cannot be totally controlled. Therefore, Tool is inversely proportional to Control (Tool = 1/Control).

Fig. 3a visualizes all the above relationships into the generalized *Compare – Contrast Model* (CCM) for robots, where the Robot Properties are depending on the Control, Autonomy, Tool, and Medium features that every robot possesses. The model suggests neither that the fluctuation rate of Control is exactly the same as the fluctuation rate of Medium, nor the fluctuation rate of Autonomy is the same with the fluctuation rate of Tool. The model implies only that their relations are proportional; if one of them increases, then the other one will increase too, but not to the same degree.

The purpose of CCM is to provide a common ground of communication -a baseline- where robot designers, developers and even users can share a mutual understanding of the potentialities, and the limitations of every robot. Thus, comparisons and contrasts between different types of robots are possible. Our interaction model has *descriptive power* (ability to describe a significant range of existing robots), *evaluative power* (ability to help assess robots), and *generative power* (the ability to help designers develop new robots). A tag on each robot with a schematic diagram that illustrates these relationships can reveal very easily its purpose, and its characteristics with just a glimpse.

V. Conclusion

We had made an attempt to present all the possible ways that could define the properties of a robot, and consequently the interactions. The expectations towards a robot functioning as a tool, are completely different from the expectations established if the robot is used as a medium of communication. The starting point for this study was a limitation the theory of HRI presented, by not having explicitly defined the notions of Direct and Indirect Interaction. Based on our observations we showcased the four main factors that affect the robot properties and the HRI, namely Control, Autonomy, Tool, and Media. The selected factors were justified by presenting a classification of robots according to them, and by explaining the reasons why we excluded the three, also important, factors of Location, Locomotion, and Appearance. Finally, we analyzed the relationships between these factors and presented the theory, the concept, the architecture, and the objectives behind the Compare-Contrast Model for robots. The proposed model aims to be used as a platform for characterizing, comparing, and contrasting robots from all the scientific areas and purposes. The CCM is easy to comprehend, and is targeting the robot designers and developers, as well as the ordinary user. Future work includes research on finding all the possible attributes of the presented factors, in order to finalize our model with a formula for the Robot Properties that would fully describe the characteristics of each robot.

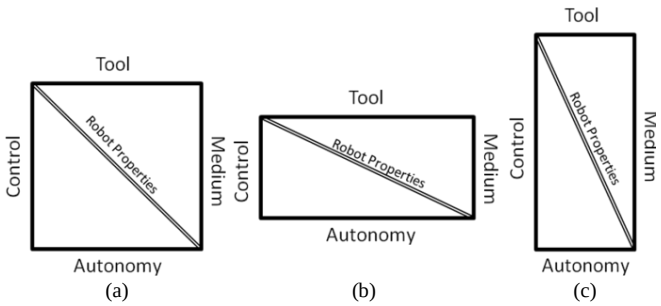


Figure 3. (a) Generic schematic diagram for the Compare-Contrast Model (CCM). (b) Variation for a robot that is more Autonomous and less Controlled, or more of a Tool than a Medium, (c) Variation for a robot that is less Autonomous and more Controlled, or more of a Medium than a Tool.

REFERENCES

[1] M. A. Goodrich, and A. C. Schulz, "Human–Robot Interaction: A Survey", Foundations and Trends in Human–Computer Interaction, Vol. 1, Issue 3, pp. 203-275, 2007, doi: http:/dx.doi.org/10.1561/1100000005

[2] J. R. Abildgaard, and H. Scharfe, "A Geminoid as Lecturer", International Conference on Social Robotics 2012 (ICSR2012), LNAI 7621, pp. 408 – 417, Springer, 2012,

[3] Harvard Biorobotics Lab. Online: http://biorobotics.harvard.edu/research.html

[4] The daVinci surgical System. Online: http://www.davincisurgery.com/davinci-surgery/davinci-surgical-system

[5] M. Y. Vardi, "Is Moore's Party Over?", Communications of the ACM, Vol. 54, Issue 11, p. 5, Nov. 2011, doi:10.1145/2018396.2018397

[6] J. Zlotowski, A. Weiss, and M. Tscheligi, "Interaction Scenarios for HRI in Public Space", Internationa Conference on Social Robotics (ICSR 2011), LNAI 7072, pp. 1–10, Springer, 2011, doi: 10.1007/978-3-642-25504-5_1

[7] K. Dautenhahn, "Methodology & Themes of Human-Robot Interaction: A Growing Research Field", International Journal of Advanced Robotic Systems, Vol. 4, Issue 1, pp. 103-108, 2007.

[8] S. Thrun, "Towards A Framework for Human-Robot Interaction", Journal Human-Computer Interaction, Vol. 19, Issue 1, pp. 9-24, June 2004, doi:10.1207/s15327051hci1901&2_2.

[9] M. Beaudouin-Lafon, "Designing Interaction, not Interfaces", Proceedings of the working conference on Advanced visual interfaces, ACM, pp. 15-22, 2004, doi: 10.1145/989863.989865.

[10] H. Kuzuoka, K. Yamazaki, A. Yamazaki, J. Kosaka, Y. Suga, and C. Heath, "Dual Ecologies of Robot as Communication Media: Thoughts on Coordinating Orientations and Projectability", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2004), Vol. 6, Issue 1, pp. 183 - 190, April 2004, doi:10.1145/985692.985716.

[11] A. Freedy, E. De Visser, G. Weltman, and N.Coeyman, "Mixed Initiative Team Performance Assessment System (MITPAS) For Training and Operation", Interservice/Industry Training, Simulation and Education Conference (I/ITSEC), Vol. 7398, pp. 1-10, 2007.

[12] E. Vlachos, and H. Scharfe, "Android Emotions Revealed", International Conference on Social Robotics 2012 (ICSR2012), LNAI 7621, pp. 408 – 417, Springer, 2012, doi: 10.1007/978-3-642-34103-8_6

[13] J. H. Ruckert, "Unity in Multiplicity: Searching for Complexity of Persona in HRI", ACM/IEE International Conference on Human Robot International 2011 (HRI'11), pp. 237 – 238, March 2011.

[14] F. Thomas and O. Johnston, The Illusion of Life: Disney Animation. Hyperion, 1981.

[15] Giraff. Online: http://www.giraff.org/?lang=en

[16] L. A. Meeden, and D. S. Blank, "Introduction to developmental robotics", Connection Science, Vol. 18, Issue. 2, pp. 93–96, June 2006.

[17] S. 17, W. Taggart, C. D. Kidd, and O. Daste, "Relational artifacts with children and elders: The complexities of cybercompanionship", Connection Science, Vol. 18, Issue 4, pp. 347–361, Dec. 2006, doi: 10.1080/09540090600868912

[18] K. Dautenhahn, B. Ogden, and T. Quick, "From embodied to socially embedded agents—implications for interaction aware robots", Cognitive Systems Research, Vol. 3, Issue 3, pp. 397–428, 2002.

[19] C. Breazeal and B. Scassellati, "How to build robots that make friends and influence people", Proceedings of the international conference on intelligent robots and systems (IROS '99), pp. 858–863, IEEE Pres,Oct. 1999

[20] M. L. Walters, D. S. Syrdal, K. Dautenhahn, R. Boekhorst, and K. L. Koay, "Avoiding the Uncanny Valley – Robot Appearance, Personality and Consistency of Behavior in an Attention-Seeking Home Scenario for a Robot Companion" , Autonomous Robots, Vol. 24, Issue 2, pp. 159-178, 2008, doi: 10.1007/s10514-007-9058-3

[21] T. Fong , I. Nourbakhsh and K. Dautenhahn, "A survey of socially interactive robots", Robotics and Autonomous Systems, Vol. 42, pp. 143–166, 2003, doi:10.1016/S0921-8890(02)00372-X.

[22] B. Duffy, "Anthropomorphism and the social robot", Robotics and Autonomous Systems, Vol. 42, Issue ¾, pp. 177–190, 2003.

[23] University of Texas robotic research group. Online: http://www.robotics.utexas.edu/rrg/learn_more/history/

[24] M. Desai, K. M. Tsui, H. A. Yanco, and C. Uhlik, "Essential Features of Telepresence Robots", Proceedings of the IEEE International Conference on Technologies for Practical Robot Applications, pp.15-20, IEEE press, April 2011.

[25] S. H. Hong, J. H. Park, K. H. Kwon, and J.W. Jeon , "A Distance Learning System for Robotics", Proceedings of the 7th international conference on Computational Science (ICCS'07), Part III: ICCS 2007 pp. 523-530, 2007, doi:10.1007/978-3-540-72588-6_85.

[26] NASA. Online: http://prime.jsc.nasa.gov/ROV/types.html

[27] A.K. Gupta, and S. K. Arora, Industrial automation and robotics, Laxmi Publications, 2007.

[28] K. Dautenhahn, and A. Billard, "Bringing up robots or - The psychology of socially intelligent robots: From theory to implementation", Proceedings of the Autonomous Agents, pp. 366-367, 1999.

[29] S. S. Ge, and M. J. Mataric, "Preface", Int J Soc Robot , Vol. 1, Issue 1, 2009, pp. 1–2, doi: 10.1007/s12369-008-0010-2.

[30] N. Epley, A. Waytz, and J. T. Cacioppo, "On Seeing Human: A Three-Factor Theory of Anthropomorphism", Psychological Review, vol. 114, Issue 4, pp. 864–886, American Psychological Association, 2007, doi: 10.1037/0033-295X.114.4.864

[31] International Standards (ISO). Online: http://www.iso.org/

[32] Anybots Virtual Presence System. Online: http://www.anybots.com/

[33] S. R. Kantelhardt, M. Finke, A. Schweikard, and A. Giese, "Evaluation of a Completely Robotized Neurosurgical Operating Microscope", Neurosurgery, Vol. 72, Issue S1, pp. A19–A26, Congress of Neurological Surgeons, 2013, doi: 10.1227/NEU.0b013e31827235f8.

[34] A. Jackson, "Brain-controlled robot grabs attention", Nature, Vol. 485, pp. 317-318, May 2012, doi:10.1038/485317a.

[35] A. Alapetite, J. P. Hansen, I. S. MacKenzie, "Demo of Gaze Controlled Flying", Proc. of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design, ACM, Oct 2012, pp. 773-774, doi:10.1145/2399016.2399140.

[36] B. Burger, I. Ferrané, F. Lerasle, G. Infantes, "Two-handed gesture recognition and fusion with speech to command a robot", Autonomous Robots, Vol. 32, Issue 2, pp. 129-147, Springer, February 2012.

[37] J. Tessadori, M. Mulas, S. Martinoia, and M. Chiappalone, "A neuro-robotic system to investigate the computational properties of neuronal assemblies," Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on , pp. 332-337, June 2012, doi: 10.1109/BioRob.2012.6290744

[38] A.J.N.V. Breemen, "Animation Engine for Believable Interactive User-Interface Robots", IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 3, pp. 2873-2878. IEEE press, 2004.

[39] R. Marin, P. J. Sanz, and A. P. D. Pobil, "The UJI Online Robot: An Education and Training Experience", Autonomous Robots, Vol. 15, pp. 283-297, Kluwer Academic Publishers, 2003.

Centralized State Estimation of Distributed Maritime Surface Oceanographers

Rasmus L. Christensen, Frederik Juul, Nick Østergaard and Jesper A. Larsen

Control and Automation, Department of Electronic Systems, Aalborg University
Fredrik Bajers Vej 7C, 9220 Aalborg, Denmark
{ralch,fjuul,nickoe,jal}@es.aau.dk

***Abstract*—This paper considers the development of a centralized controller for the purpose of navigating a small Autonomous Surface Vehicle (ASV). The centralized controller is using a Kalman filter as a state predictor to improve the precision of the navigational aids mounted aboard. This work presents the design of the motion control system as well as an estimator designed to cope with packet losses.**

***Keywords*—Centralized control; baud rates; state estimation; marine systems; master slave system.**

I. Introduction

Seaborne measurements are often an expensive and time-consuming task. They could however in many cases have a large impact on the area where they are obtained. At the Fukushima accident in 2011 the area of effect in the water and the safety margin around it was primarily based on estimates, as only a few measurements were available, and the risk of sending people into radiated areas was considered too large.[1].

Another area the could benefit, is the coast around Greenland, as up to date maps are currently not available. This causes the ships which need to pass near the coast to have a higher safety margin, which in turn lowers the traffic throughput. If up to date maps were available, the risk of a cruise ship running aground would be lowered. This need is acknowledged and is considered important for the flourishing Greenlandic industrialization by the government of Denmark, but the project does not have sufficient funding. [2].

One way to reduce the cost of such maritime measurements would be to develop small autonomous drones to carry out the task. These should be controlled by a mother ship to enable easy scalability and coordination. Further, they should communicate using a simple data link to preserve bandwidth and limit power consumption. The system should be robust to dataloss and measurementnoise which can be expected from low cost sensors. This can be achieved through sensor fusion.

Currently the main focus of autonomous vehicles have been on aerial, ground and underwater vehicles, while there is close to no research going on about small autonomous surface vessels. An example of such a vessel is the Stingray USV developed by Isreali based Elbit Systems.

Figure 1 depicts the Stingray. It is primarily developed as an aid in the battle agains pirates and for SAR (search and rescue) missions, but can also be equipped with other sensory equipment, which can be used to map the sea bed, or measure the amount of radiation in coastal areas.

Fig. 1. The Unamnned Surface Vehicle (USV) Stingray by Israeli based Elbit Systems in action [3].

The scope of this project is however to develop a smaller vehicle, which in turn could function as part of a measuring swarm, controlled by a mother ship, thus making measuring missions more efficient and less time consuming. As the price of such a swarm system for measurement purposes would be high if systems like the Stingray is to be used, the suggestion of this project, is to develop a small cost-efficient vessel. Throughout this project, a prototype vessel AAUSHIP.01 have been developed, taking into account the hull design, choosing electronics and so forth.

This paper will only describe the development of the estimation and control algorithms. Figure 2 depicts the prototype used in the project. This ship is fitted with two 1200W engines which are run as a differential system (making it a ship without rudders), making the ship turn by reducing the input on one engine and increasing it on the other, and vice-versa.

A. Problem Statement

Is it possible to develop a centralized state estimator for use in the maritime environment using a small data link?

II. Methods

This project is primarily based around the linearised state space model derived for the ship, which are then utilized as

Fig. 2. AAUSHIP.01—a prototype of an autonomous maritime surface oceanographer, as this is only a prototype, the lid is held in place with clamps.

a reference for developing the control strategy used in the project. The notations in the report is based on the SNAME [4] notation.

A. Physical Modeling of the Ship

As Aalborg University does currently not own any small maritime vessels, such a ship would have to be developed. A quick draft of the onboard electronics quickly sketched the outlines and dimensions of the project, and by using 3D modeling techniques the design depicted on figure 3 was reached.

The ship is designed as a non-planing displacement hull. This design is primarily used on cargo vessels, as these travel at a low speed and are able to carry big payloads, allowing for AAUSHIP.01 to carry a big payload, whilst sailing at a low enough speed to carry out measurements. The ship is equipped with two 1.2 kW engines to provide the thrust needed to propel the ship along (even though it might experience strong down-stream currents from glacial rivers), which are run differentially, providing torque by reducing the input to one engine and increasing it on the other.

Initial tests of these powerful engines, proved the ship to start planing even at low speeds and initiated ventilation of the propellers, thus reducing the thrust [5]. To counter this, an aft hydrofoil was mounted, and the pitch of the propellers were inversed to avoid ventilation and reduce pitch and planning. Tests proved this to increase stability in pitch, even at high cruising speeds.

The ship can be retrofitted with a tunnel thruster to help tight quarter navigation if this is required at low speeds, but this has not been implemented on the prototype.

B. Mathematical Modeling of the Ship

AAUSHIP.01 is designed as a non-planing displacement ship, which are not to sail in any form of rough sea, which makes for the simplification of the vessel being modeled as a 2 degree of freedom vessel, with the states given as surge, yaw and yaw rate $[x, \psi, r]^T$, describing the motion in the x-direction, the angle about the z-axis ψ and the rotational velocity about the z-axis r.

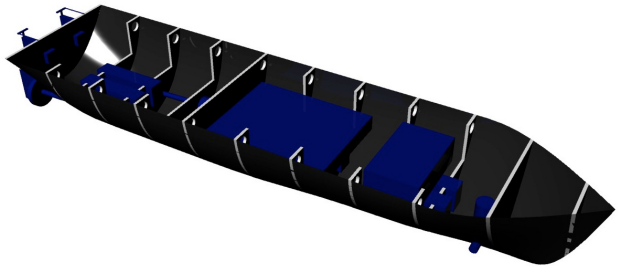


Fig. 3. AAUSHIP.01 – 3D rendering of the prototype hull, before being produced. The blue boxes indicate space for on-board computers and other hardware.

$$\begin{bmatrix} \dot{u} \\ \dot{\psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\beta_{\dot{x}} & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\beta_{\psi} \end{bmatrix} \begin{bmatrix} u \\ \psi \\ r \end{bmatrix} + \begin{bmatrix} m^{-1} & 0 \\ 0 & 0 \\ 0 & I^{-1} \end{bmatrix} \begin{bmatrix} X \\ N \end{bmatrix} \quad (1)$$

In (1) β denotes the skin frictional drag resistance divided by the mass m ($\beta_{\dot{x}}$) and the inertia I ($\beta_{\dot{\psi}}$) respectively. This equation is of course a simplified version of the entire system, where the effects of damping, coriolis and the like have not been accounted for. This simplification is fair as AAUSHIP.01 is relatively small, and the areas it is built to map are small as well. For the purpose of controlling the vessel, an LQR feedback controller have been implemented alongside a reference tracking gain, these are described in (16) and (17). The system is implemented on the on-board controller, and is discretised using zero-order hold.

C. Navigation

The path planner computes the reference heading for the heading controller. In order to efficiently and accurately navigate along the path, a set of sub-waypoints is calculated for each path segment between two waypoints, these sub-waypoints are then used as a navigational aid. The main control strategy is to navigate through all of these sub-waypoints in a predefined order, one by one. The heading of the ship is defined using the NED[1] frame[6]. The required heading is determined based on the position and heading of the ship and the position of the next sub-waypoint. This topic it not the focus of this paper and will not be described further.

D. State Estimation

To give a better estimate of position and the attitude of the vessel, a Kalman filter have been implemented to improve the accuracy of the sensors mounted aboard the ship. To develop a such, the discrete time state model of the ship have been derived to be:

$$\mathbf{\Phi} = \text{diag}\{\mathbf{\Phi}_x, \mathbf{\Phi}_y, \mathbf{\Phi}_\psi\} \quad (2)$$

Where the diagonal entries $\mathbf{\Phi}_x, \mathbf{\Phi}_y, \mathbf{\Phi}_\psi$ are given by the same equation, with different entries:

$$\mathbf{\Phi}_{x,y,\psi}(k) = \begin{bmatrix} 1 & t_s & 0 \\ 0 & 1 & t_s \\ 0 & -\beta_{x,y,\psi} & 0 \end{bmatrix} \quad (3)$$

[1] [North, East, Down]

In equation (3), $\beta_{x,y,\psi}$ denotes the skin frictional drag resistance in the x, y and ψ direction respectively, m is the mass of the vessel, I is the inertia and t_s is the sampling time of the filter. The acceleration of the vessel $\dot{u}$, $\dot{v}$ and $\dot{r}$ are omitted as the error grows exponentially (due to a linear filter trying to estimate a non-linear problem). The states to be estimated for the controller are thus:

$$^b\hat{\boldsymbol{x}}_{\mathbf{k}} = \begin{bmatrix} x & u & y & v & \psi & r \end{bmatrix}^T \tag{4}$$

The onboard sensory equipment are able to output the following measurement vector $\boldsymbol{v}_l$:

$$\boldsymbol{v}_k = \begin{bmatrix} x & u & \dot{u} & y & v & \dot{v} & \psi & r & \dot{r} \end{bmatrix}^T \tag{5}$$

To tune the filter, the covariance matrices $\boldsymbol{R}_k$ and $\boldsymbol{Q}_k$ have to be tuned. These are function of the measurement distribution and the input distribution. The input to the system is given as:

$$\boldsymbol{w}_k = \boldsymbol{B}\boldsymbol{u} \tag{6}$$

Where $\boldsymbol{B}$ is an augmented version of the system input matrix defined in (1) and $\boldsymbol{u}$ is the inputs to the system given as a force X and a moment N. $\boldsymbol{B}$ is given as

$$\boldsymbol{B} = \begin{bmatrix} 0 & 0 & \frac{1}{m} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{I} \end{bmatrix}^T \tag{7}$$

And the input is given as $\boldsymbol{u} = \begin{bmatrix} X & N \end{bmatrix}^T$. As $\boldsymbol{B}$ is a static matrix, it is only the distribution of the force F and the torque τ that is of interest. The distribution of the force affecting the ship can be split into an x- and y-component, as the ship is also expected to move slightly sideways. The distributions of the force is given as white Guassian noise processes, thus stating:

$$X \sim \mathcal{N}(\boldsymbol{\mu}_X, \boldsymbol{\sigma}_X^2) \tag{8}$$

$$N \sim \mathcal{N}(\mu_N, \sigma_N^2) \tag{9}$$

Where the tuning through simulations have yielded the best results using $\boldsymbol{\mu}_X = \begin{bmatrix} 5.4355 & 0 \end{bmatrix}^T$. The first entry is because to the ship is for most of the time moving along at 1 m/s and that is the estimated force required to thrust the ship forward at 1 m/s with the frictional drag the ship experiences. The latter is the force in the y-direction which is given as a zero-mean process as the ship generally is expected not to move sideways. The torque is also given as a zero-mean process as the ship for most of the time is moving straight. The covariance matrix for the input, can thus be given as the covariance of $\boldsymbol{B}\boldsymbol{u}$ which gives the following:

$$\boldsymbol{R}_k = \text{diag}\{0, 0, \sigma_{X(1,1)}^2, 0, 0, \sigma_{X(2,1)}^2, 0, 0, \sigma_N^2\} \tag{10}$$

The measurement covariance is more interesting, as this contains the actual variance of the measurements, and play a big part in how much each of these are weighted in Kalman Filter $\bar{\boldsymbol{K}}$. Static tests have been conducted to estimate these variances, and all the sensors are assumed to be Gaussian white noise processes, thus defining:

$$\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{\mu}_v, \boldsymbol{\sigma}_v^2) \tag{11}$$

As all the measurements are independent, the covariance matrix is given as a diagonal matrix:

$$\boldsymbol{Q}_k = \text{diag}\{\boldsymbol{\sigma}_v^2\} \tag{12}$$

Where the individual entries of this is defined to be:

$$\boldsymbol{\sigma}_v^2 = \begin{bmatrix} \sigma_x^2 & \sigma_u^2 & \sigma_{\dot{u}}^2 & \sigma_y^2 & \sigma_v^2 & \sigma_{\dot{v}}^2 & \sigma_\psi^2 & \sigma_r^2 & \sigma_{\dot{r}}^2 \end{bmatrix}^T \tag{13}$$

As the velocity output of the GPS device $\dot{x}$ is an absolute value, the distribution of this is a bit different from the other sensors. To estimate the variance of $\dot{x}$ the unbiased sample variance formula is used, which computes the variance, even though the samples only consist of absolute values.

The actual implementation of the filter is an altered version of an Linear Minimum Mean Square Error filter – the alteration lies in the Kalman gain, where a matrix mask $\boldsymbol{\Lambda}$ is post multiplied. This matrix mask is to zero out the measurements that are invalid. This matrix mask is defined as:

$$\boldsymbol{\Lambda} = \text{diag}\{\lambda_x, \lambda_u, \lambda_{\dot{u}}, \lambda_y, \lambda_v, \lambda_{\dot{v}}, \lambda_\psi, \lambda_r, \lambda_{\dot{r}}\} \tag{14}$$

This ensures that when a measurement is invalid (the checksum is not true) the receiver zeros out the gain, and runs the filter on the other estimates. The individual λs are thus given as a function of the checksum:

$$\lambda = \begin{cases} 1 & \text{if checksum is valid} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

This makes sure to zero out the Kalman gain $\bar{\boldsymbol{K}}$ if a packet is corrupted instead of making the filter run on faulty data. When this is implemented, the system handles packet loss by making the filter run on estimates for the next sample, rather than running on a faulty measurement.

E. Modelling and Controls

The model of the ASV, can in continuos time be given as the state space equation defined in (1) – the model considers the motion of the ship with 2 degrees of freedom, movement in the x-direction, angle about the z-axis ψ and rotational velocity about the z-axis r. This reduced model is sufficient as it is only the velocity and the angle that are controllable in the current configuration.

The control strategy for this project is to track the input reference, and use an optimal feedback gain to reach the desired values and a reference gain used to zero out the steady state error. This is done as in [7] resulting in the following gain matrices:

$$F_{\text{opt}} = \begin{bmatrix} 15.1668 & 0 & 0 \\ 0 & 2.5165 & 0.7134 \end{bmatrix} \tag{16}$$

$$N_{\text{ref}} = \begin{bmatrix} 24.0668 & 0 \\ 0 & 2.5165 \end{bmatrix} \tag{17}$$

For implementation purposes the system is discretised using the MATLAB command `c2d` with a samping time of $t_s = 1/3$ (the rate at which measurements are available from the GPS), and then using the same algorithms as in equation (16) and (17) for computing the optimal feedback and reference gain respectively, thus resulting in the gains (18) and (19):

$$_{\mathcal{D}}F_{\text{opt}} = \begin{bmatrix} 10.6956 & 0 & 0 \\ 0 & 2.2743 & 0.6681 \end{bmatrix} \tag{18}$$

$$_{\mathcal{D}}N_{\text{ref}} = \begin{bmatrix} 19.5956 & 0 \\ 0 & 2.2743 \end{bmatrix} \tag{19}$$

As the controller outputs a desired force X and a torque N dependent on the reference angle ψ and velocity u. This output

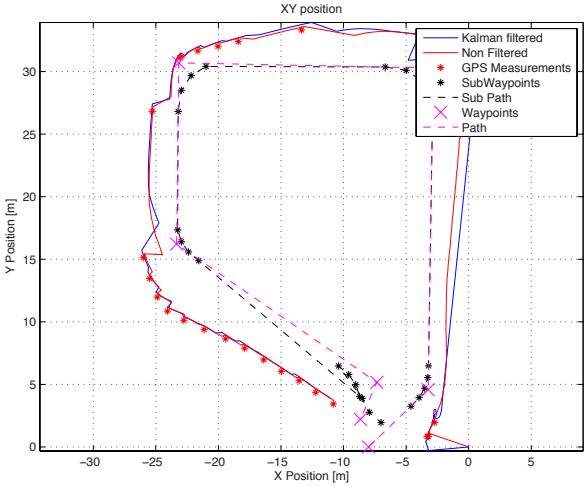


Fig. 4. Test 1 of the control and position estimation. The test was performed in the Klingenberg Pond at Aalborg University. As seen on the figure, the ship looses GPS signals due to the surroundings, but are still able to navigate the track.

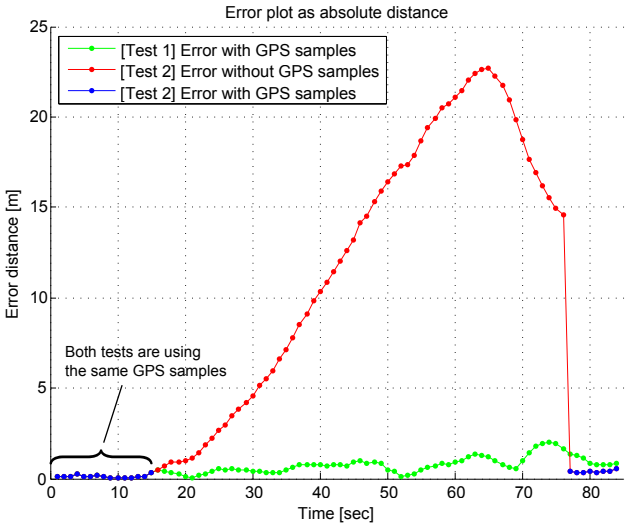


Fig. 5. Error between measured and actual path from figure 8. Two "tests" were performed, where test 1 is when the filter looses no GPS samples. Test 2 is with GPS samples missing for about 60 seconds, this increases the error, but the vessel is still able to navigate using the Kalman filter.

must be converted to a set-point revolution, as an input to the low-level system handling the engines. The force of the forward motion of the ship is given as a function of the number of revolutions of the propeller n, this relation is defined by [8] to be as in (20):

$$X = \rho \cdot D^4 \cdot K_T \cdot |n| \cdot n \tag{20}$$

As the two engines both produce both a force and a torque, as a function of the revolution vector $[n_1, n_2]$, the matrix equation $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$ can be solved for the number of revolutions the engines need to produce as in (21):

$$\begin{bmatrix} n_1^2 \\ n_2^2 \end{bmatrix} = \begin{bmatrix} C_1 & C_1 \\ C_1 \cdot l \cdot \sin(\psi_{\text{stbd.}}) & C_1 \cdot l \cdot \sin(\psi_{\text{port}}) \end{bmatrix}^{-1} \cdot \begin{bmatrix} X_{\text{desired}} \\ N_{\text{desired}} \end{bmatrix} \tag{21}$$

The revolutions of the propeller can assume negative values (if the ship is reversing), so what is left to do now, is to solve for n_1 and n_2. These can be solved by rearranging (21), such that the setpoints is calculated as:

$$n_{\text{port}} = \text{sign}(n_1^2) \cdot \sqrt{|n_1^2|} \tag{22}$$

$$n_{\text{stbd.}} = \text{sign}(n_2^2) \cdot \sqrt{|n_2^2|} \tag{23}$$

Thus giving the engine setpoints as a function of the desired forward force X and torque N of the system.

F. Controller Verification

To verify wether the system actually works, tests have been conducted in a small lake at the Klingenberg Lake at the Aalborg University campus area. The goal of the test, was to verify that the theory used in developing AAUSHIP.01 actually worked. The test was performed in calm water, and the ship was programmed to follow a path defined by the black asteriks (*) on figure (4).

The reason for the ship not converging towards the track on the straights line segments is due to a software bug, that could not be solved at the current time. The algorithem has worked for the same kind of scenarios in simulations as the field test. Therefore the only place where the ship actually converges properly towards the track is in the corners.

To verify that the Kalman filter improves the estimate and thus produces better state measurements to the system, 6 is a figure representing the different states of the system, as the ship traverses the route depicted on figure 4. The error plot have been produced to verify that the controller actually acts on the input. On figure 8 and 5 it is seen that the estimator still produces an estimate of the actual position, while this is off, it is still valid as a position estimate (better than sailing blindly). The error converges to near zero as soon as the ship receives a new valid GPS measurement.

G. System Verification

To ensure that the track sailed on figure 4 was not just a one of, the system was run multiple times with the same track as a reference. This produced as shown on figure 7. On the figure it is clearly seen that the ship consistently follows the contours of the track, but the estimates sometimes wander of due to loss of GPS reception.

H. Errors

As seen on figure 5, 7 and 8 the estimates wander off when the ship navigates in areas of bad GPS reception, this is due to the unknown and uncompensated bias of the IMU measuremts, which will cause fast divergence under dead-reckoning.

I. Discussion

To further improve on the system, the Kalman filter should be extended to allow for the acceleration data also to be included in the position estimates, thus increasing the precision

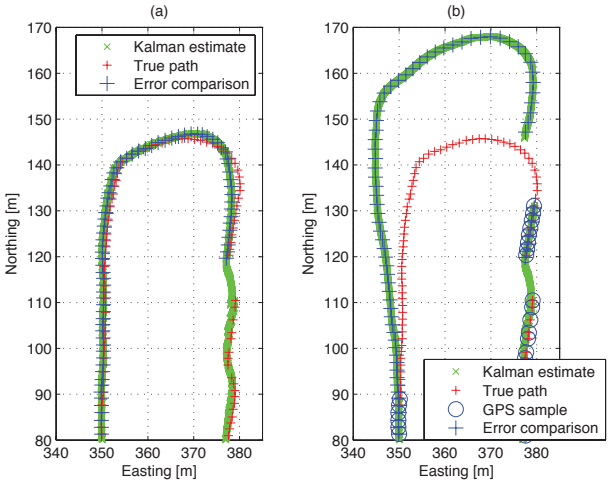


Fig. 6. Measured and estimated position of the vessel, during a 180 degree turn (on shore). This test was carried out to validate the performance of the Kalman filter, before implementing the design on the vessel.

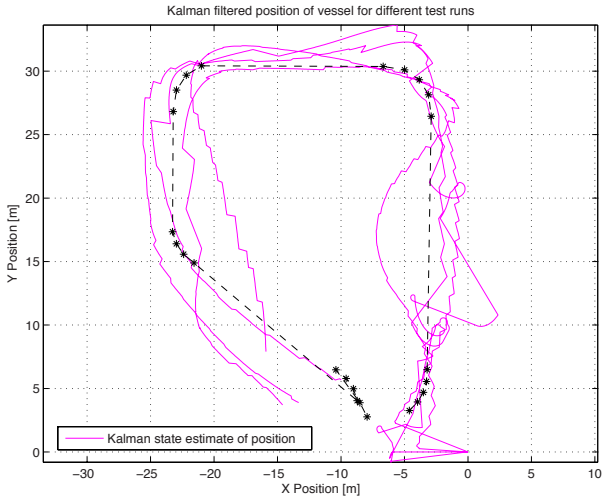


Fig. 7. System verification with multiple tests run and plotted on top of the track. As seen the vessel consistently follows the contours of the track, whilst at some points loosing some precision (due to the nature of the surroundings, not allowing for optimal GPS reception).

further when the GPS is offline. To make this feasible the bias of the IMU should be included in the model. Another feature of the Kalman filter would be to estimate the environmental forces and moments, which would cause the ship to veer off course.

Throughout the tests the IMU have been assumed to always be level, which is hard to realise, so a slight bias is added, and with a linear Kalman filter which is hard compensate for as the IMU needs to always have the same bias for this to work.

III. Conclusion

To sum up the project, a state estimator have been developed to improve on the GPS estimates allowing for a better precision if the GPS is offline (dead-reckoning). To make

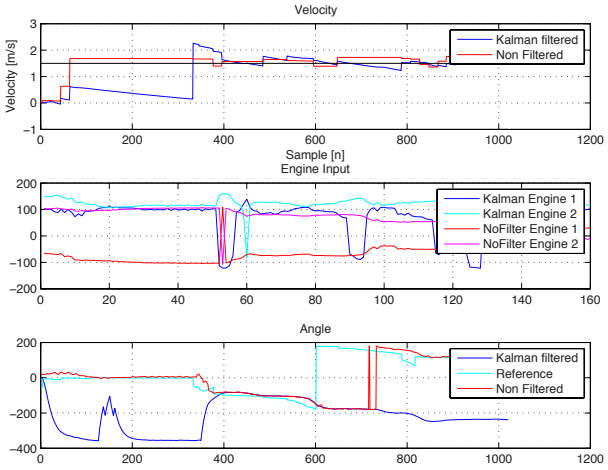


Fig. 8. Measured and estimated states of the vessel. As seen on the velocity plot, the ship tracks the velocity. The reason for the Kalman filter not being able to track the velocity for the 330 samples, is due to the bad GPS reception. The Engine input plot, is however a lot better and smoother for the estimated states, whereas the measured states would have the ship sail in circles for the first period of time. The last figure is the heading (do note that the angles might be 360 degrees of).

the ship traverse a desired path, a path planner has been implemented to guide the low level contol such that path folowing can be achieved in an sensible way.

A final test of the combined system have not been conducted, as the waters around Aalborg have been frozen solid.

Acknowledgment

A special thank should be given to Assistant Professor Carles Navarro Manchón, Section for Navigation and Communication, Department of Electronic Systems, Aalborg University for his help with tuning the Kalman filter and to School of Information and Communication Technlogy for the donation without which this project would not have succeded.

References

[1] U.S. Department of Energy, *Radiological Assesment or effects from Fukushim Daiichi Nuclear Power Plant* U.S. Department of Energy

[2] Birgitte Marfelt, Ingeniøren *Tom pengekasse bremser søkort omkring Grønland* Mediehuset Ingeniøren, 2012

[3] Defense Update, 2005, Issue 2, *Stingray, Unmanned Surface Vehcile (USV), Elbit Systems* Defense Update

[4] Technical and Research Bulletin No. 1-5, *Nomenclature for Treating the Motion of a Submerged Body Through a Fluid* SNAME

[5] Foiltheory, Lecture Notes, 2012, *Lecture Notes on Foiltheory* MARINTEK

[6] Thor I. Fossen, 2005, *Handbook of Marine Craft Hydrodynamics and Motion Control* Wiley

[7] Gene F. Franklin, J. David Powell and Abbas Emami-Naeini, *Feedback Control of Dynamic Systems, 5th ed.* Pearson Prentice Hall, 2009

[8] Asgeir J. Sørensen, *Marine Cybernetics, Modelling and Control – Lecture Notes* Department of Marine Cybernetics, Norwegian University of Science and Technology, 2005

DESIGN- AND IMPLEMENTATION OF TELE-SURGERY ROBOT, WITH FORCE FEEDBACK.

Thomas Hansen, Claus T. Henningsen, Rasmus Pedersen,
John Schwensen, Senthuran Sivabalan, Jesper A. Larsen and John Leth

Control and Automation, Department of Electronic Systems, Aalborg University
Fredrik Bajers Vej 7C, 9220 Aalborg, Denmark
{11gr733, jal, jjl}@es.aau.dk

***Abstract*—Surgical robots have become increasingly used as a tool for performing certain abdominal procedures without requiring open surgery. This form of minimally invasive surgery has the advantages that it can be performed with less trauma to the patient and the surgeon can be seated in a more comfortable position. However, this currently comes at the cost of complete loss of touch and the surgeon must instead rely on visual cues. In this paper a prototype for a haptic feedback system is presented that will reenable the sense of touch through the remotely placed joystick. This system is able to estimate the forces exerted by the end-effector through modelling of the implemented motors and available measurements. Tests show this is indeed possible and haptic feedback can be applied even with considerable delay between joystick and end-effector.**

***Keywords*—Force-Feedback, Tele-surgery, Parameter Estimation.**

I. INTRODUCTION

In certain environments or situations where humans are not capable of performing tasks, remotely controlled robots are commonly used in today's society. In the field of surgery, remotely controlled robots have been in use for some time. Typically in such a setup the surgical robot end-effectors and cameras are the only things inside the patient. The surgeon can then perform the operation remotely by using a set of control handles and by visual feedback from the cameras. The small size of cameras and end-effectors makes minimal invasive surgery possible, which puts much less stress on the patient, reduces scarring and also minimizes the chance for post operational infection due to the relatively small incisions[1]. This also means that the patient recovers faster which is highly beneficial for the society, as hospitalization is expensive. However, the surgeon is via the surgical robot able to apply pressure only with the visual confirmation of his actions. By giving the surgeon, the ability to feel the impact of his instruments in the remote operating environment could improve the perception of the reality at and in the patient. The added sense could help improve the quality and speed of surgery by combining more of the surgeons senses with the benefits of the steady robotic hands and the minimal invasive endoscopic surgery. There are different methods of simulating the sense of touch at the joystick depending on the hardware, complexity and/or lifelikeness. An example of two such methods could be either the simple one utilized among other places in many joysticks intended for games and nearly all cell phones, where a small motor with an offset mass on the rotating axis can create a rumbling sensation of varying intensity. The other method is able to give a more realistic feeling by being able to counter the motion of the users hand with an adjustable force and direction. This is also known as force feedback.

The advantages of adding the sense of touch to a telesurgical system can basically be described as countering the major disadvantage that telesurgery introduces in the first place. For surgeons new to the system the learning curve can be very steep as they have to get used to losing a critical sense, and even experienced users could possibly improve by having the ability to feel what they are doing as when doing normal surgery.

The concept of telesurgery is one that has been researched for many years and has resulted in the world's first transatlantic surgery in 2001 by [1], with the surgeon sitting in an office in New York and performing a cholecystectomy[2] on a woman in Strasbourg. This was done over a dedicated fiber-optic connection with a mean round trip time of 155 ms. Further studies into the effects of time-delay in a telesurgical environment has been done such as [2], who concludes that delays of 400 ms or below had no significant impact on neither task time nor error rate when performing surgical like tasks. [3] have studied the combination of time-delay and force-feedback concluding that in a suturing task the skill of the surgeon outweighed the negative effects of 165 and 270 ms delay. Neither did force feedback increase completion time for experienced surgeons, however it did significantly reduce forces exerted upon the sutured fabric. Both of these studies confirm [4] who concludes that time-delays of 600 ms significantly increases task times. In addition they also show that asynchrony in video and control feedback improves performance compared to further delaying one of the signals to obtain synchronised feedback.

This paper describes the work done to design and implement a robotic surgery system in relation to tele-surgery. The solution also includes an implementation of force feedback. The design includes precautions discovered in the analysis phase of the project, however only core functionalities were included in the implementation. A prototype is constructed, and is tested. The test are conducted on local networks with simulated reproducible network delays.

In Section II, the design, the prototype and the environment will be further described along with any definitions and assumptions made during the design of the system. Further-

[1]This claim is based on information given by nurses and surgeon from Aalborg University Hospital.

[2]Removal of the gallbladder.

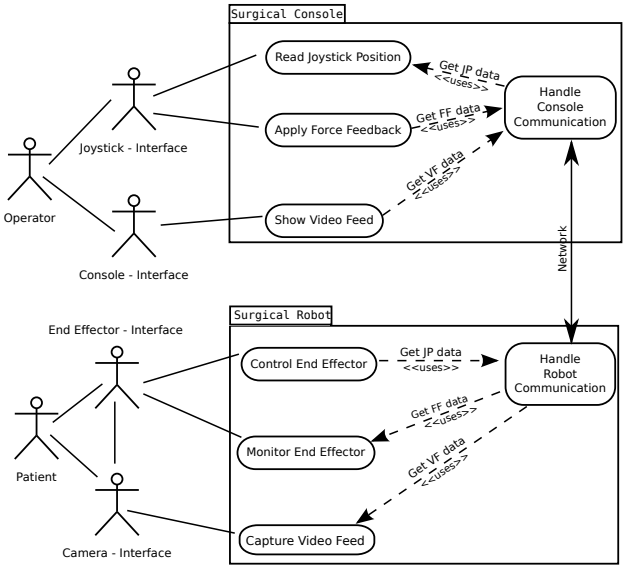


Figure 1: *The use case diagram shows the core functionalities of the system. The Systems, Actors and Use cases are shown in the diagram. Here JP is Joystick Position, FF is Force Feedback and VF is Video Feed.*

more, the ideas and methods behind both control design and the development of the parameter estimator will be described before introducing the practical tests, through which the results have been obtained. Section III, will take a closer look upon the details of the data gathered in the specified tests of the system. Lastly, Section IV, will conclude on the obtained results while Section V, will leave suggestions regarding areas for further research and development.

II. METHODS

A Design

When designing the surgical robot, an extremely important issue is safety. Therefore there is needed to be implemented safety protocols ensuring that errors, of human and technical origin, will affect the patient as little as possible.

A.1 Functionality

To identify the cases where errors could emerge, the use case diagram shown in Figure 1 This use case diagram is used as a framework for the project, as it describes the core functionalities of the product. However the use case diagram does not limit the project, and there is still opportunity for expanding the project. The use cases are divided into two subgroups, given by the system that contains them. These subgroups are *Surgical Console* and *Surgical Robot* and the most critical use cases and security measures taken are described here:

Surgical Console - Read Joystick Position

Normal scenario:

1. *Operator:* Moves the Joystick.
2. *Surgical Console:* Reads the current position of the *Joystick*.

Security measures:

- Joystick position is not available for the surgical console.
  The Operator gets an error indicating that the position is unavailable, and the read position is set to N/A.
- Joystick position is exceeding the limit values set for the Joystick Interface.
  The Operator gets an error indicating that the limits are exceeded, and the read position is set to the maximum allowed value in that direction.

Surgical Console - Apply Force Feedback

Normal scenario:

1. *Surgical Console:* Retrieve FF data by using the *Handle Console Communication* use case.
2. *Surgical Console:* Apply actual transformed FF to Joystick.

Security measures:

- The FF data is unavailable or corrupt.
  The Operator gets an error indicating that the FF data is unavailable, and the applied FF is set to the FF received in the previous data, if it exists, otherwise no FF is applied.
- The FF corresponds to a force, applied by the End-effector, which exceeds the limit for force allowed.
  The Operator gets an error indicating that the force-limit is exceeded, and FF corresponding to the limit is applied.

Surgical Robot - Control End-effector

Normal scenario:

1. *Surgical Console:* Retrieve FF data by using the *Handle Console Communication* use case.
2. *Surgical Console:* Apply actual transformed FF to Joystick.

Security measures:

- JP data is unavailable.
  Keep End-effectors in the same position.
- *End-effectors* are defect.
  The End-effectors are kept in the same position.

Surgical Robot - Monitor End-effector

Normal scenario:

1. *Surgical Robot:* Monitor force applied, which is measured by the *End-effectors*.

Security measures:

- Unable to read force.
  Set FF data to N/A.
- Failure Detected in *End-effectors*.
  Set FF data to indicate error on console.

A.2 Network

When performing surgery over longer distances, also known as telesurgery, a critical component of the system is the network used to connect the console with the operating robot. The two dominating factors are the bandwidth of the connection between the two stations, and the time delay introduced by the connection. Where the bandwidth can limit the amount of information sent between the stations, the time delay can impact on the surgeons perception of the procedure.

Dedicated links between locations could be a solution to ensure a connection, as seen on the da Vinci Surgical System [6]. The two systems are connected with a cable made for that purpose, capable of handling the required transmission. However in general it could be advantageous to instead use an Internet connection, if it is possible to meet demands in both bandwidth and time delay.

The bandwidth requirement to the connection can be calculated from the known amount of information needed to

be sent, which should be well defined during design of the system. Time delay requirements on the other hand can be more difficult to define. While longer delays might not affect the workings of the system itself it could have catastrophic effects on the surgeon's perception of what goes on in the patient, and in the end the success of the operation. Studies show that delays in the range of 0-400 ms, from input to visual feedback, have no significant impact on either error rate or speed of tasks [12]. However any larger delay will result in both significantly more time used and errors made. These results are supported by the previously used study that concludes delays of 270 ms did not strongly impact procedures [13]. This study furthermore concludes that the skills of the operator is what compensates for the delay since inexperienced operators were around 50% slower on task-completion.

It is decided to use the Internet (Ethernet connected to the Internet) as network for this project. This is mainly done for the purpose of examining whether or not it is actually possible and what considerations should be taken when using it. Due to the availability of it, it could be an easy solution for making it possible to perform telesurgery using existing networks instead of investing in a whole new infrastructure.

The packets sent within the system are mainly of two types, video- and control packets. The video module have to send approximately 20 kB, between 10 and 25 times/s. depending on the network traffic. This gives a required maximum throughput of 500 kB/s. The control modules have two send position, velocity, acceleration and force feedback info from each joint. A robot arm with end-effector consists of up to 10 joints and it is estimated that 4 robot arms will be sufficient for now. It is estimated that 4 bytes will be sufficient for each parameter and the control loop packets need to be sent up to 1000 times/s [11, p. 194]. Then the control loop packet size becomes:

$$\text{Packet Size} = 4\,\text{Info} \cdot 10\,\text{Joint} \cdot 4\,\text{Arm} \cdot 4\,\text{bytes} = 640\,\text{bytes} \tag{1}$$

Whilst the estimated packet size the required throughput becomes:

$$\text{Throughput} = 640\,\text{bytes} \cdot 1000\,\text{s}^{-1} = 640\,\text{kB/s} \tag{2}$$

As it is possible to make the network run in full duplex mode, the highest required throughput is from the robot to the console, where both video feed and control loop packets need to be sent. The maximum required throughput thereby becomes:

$$\text{Max Required Throughput} = 500\,\text{kB/s} + 640\,\text{kB/s} = 1140\,\text{kB/s} \tag{3}$$

This gives a bit rate of 9.12 Mbit/s, excluding any overhead required for data transfer. It is deemed realistic that an Internet connection could support this.

B Implementation

The implementation of the remote surgical system is done by constructing a modular and easily expandable system, to ensure future modification and feature adding possible. Therefore interfaces between subsystems have been defined, i.e the interface between the workstations and the motors are processed by separate motor drivers. The hardware setup is shown in Figure 2.

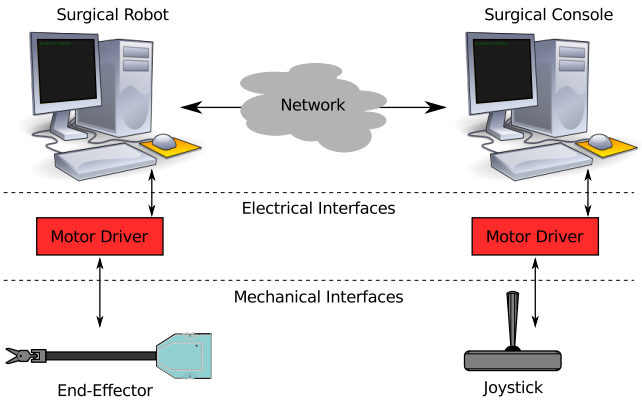


Figure 2: *Setup of the developed remote surgical system prototype.*

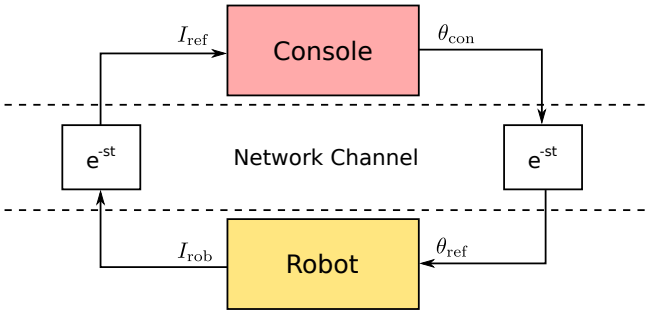


Figure 3: *Concept overview of the control system.*

The two computers are running a low latency Linux kernel, which has a considerably lower latency than the corresponding generic kernel [5], but still do not apply hard real time. The motors, applying force feedback to the joystick and driving the end-effector, are handled by motor drivers, implemented as embedded systems, through a serial link. The end-effector used for the prototype system is a part taken from the da Vinci Surgical Systems [6] and has four degrees of freedom. A corresponding joystick has been developed, also with four degrees of freedom by Aalborg University.

The link between the workstations, consist of an Ethernet communication channel. This is utilised by using the User Datagram Protocol (UDP) of the Internet Protocol Suite.

B.1 Control

The control design are implemented as two systems, being the surgical console and the surgical robot. An overview of the control design is shown in Figure 3. The objective of the surgical consoles control design is ensuring the force applied by the end-effector is fed back, unscaled through the joystick, to the user. Meanwhile the control design of the surgical robot, ensure the position of the joystick is matched by the end-effector. Each plant model has one input and two outputs as shown in Figure 4 on the next page. This is done such that it is possible to control the internal loops and give reference points to the corresponding control loop. The feedback from the surgical robot to the surgical console is negated, in order to counter act the movement of the end-effector and hand.

The two inner loops, closed loop transfer functions, are stated in Eq. 4 on the following page and 5 on the next page. H_{c} is the surgical consoles closed loop and H_{r} is the surgiccal

robots closed loop.

$$H_c = \frac{C_c \cdot G_{1c}}{1 + C_c \cdot G_{2c}} \tag{4}$$

$$H_r = \frac{-C_r \cdot G_{2r}}{1 + C_r \cdot G_{1r}} \tag{5}$$

To analyse the stability of the system an open loop transfer function is used, derived in Eq. 6

$$H_{ol} = H_c \cdot H_r \cdot e^{-2st} \tag{6}$$

To ensure a stable system considerations are made to all the transfer functions originating from the multiple inputs and outputs of each subsystem. By using the principle of internal stability, the surgical console and surgical robot models have been individually analysed for stability across all of their transfer function, by checking the poles of the system. Hereafter by considering a single transfer function through the system, the complete system can be proven stable [8]. This transfer function, $\frac{I_{rob}}{I_{ref}}$, is used to determine the effects of time-delay.

With the designed controllers the system is stable, however the phase margin is as low as 5°. As this makes it possible for model inacuracies to push the system into instability, a lead compensator is designed in series with the main loop. Furthermore Smith predictors are implemented to suppress the destabilizing effects of the pure time delay. The smmith predictor is implemented as a model of the surgical robot. Both the lead compensator and the Smith predictor are shown on Figure 5. The smith predictor utilises the roundtrip time of the system, which must be provided to it. And it is therefore implemented into the protocol, that the roundtriptime is measured.

B.2 Parameter Estimation

The current through the DC-motors are used to calculate the torque of them, as this is directly proportional. To avoid adding sensors to the system, an estimation of the torque, based on the position sensor is used. This sensor is used as it is a crusial part of any robotic surgery solution, and is therefore deemed most likely to be available in most robotic surgery systems. When the position and the sampling frequency is known, it is trivial to calculate the velocity.

A state space DC-motor model is found with the states being the current, i, and the angular velocity, ω, and the input being the motor voltage, u. The model in Eq. 7 and 8 is in

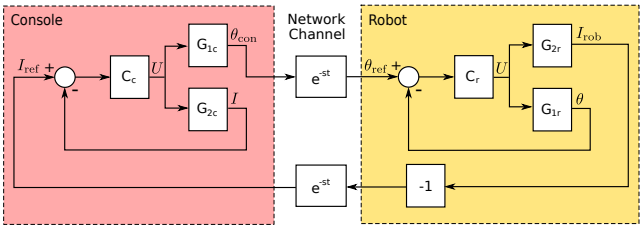


Figure 4: *Expanded block diagram of the control system, with the different transfer functions. C_c is the console controller and C_r is the robot controller. $G_{1c,r} = \frac{\theta}{U}$, $G_{2c,r} = \frac{I}{U}$.*

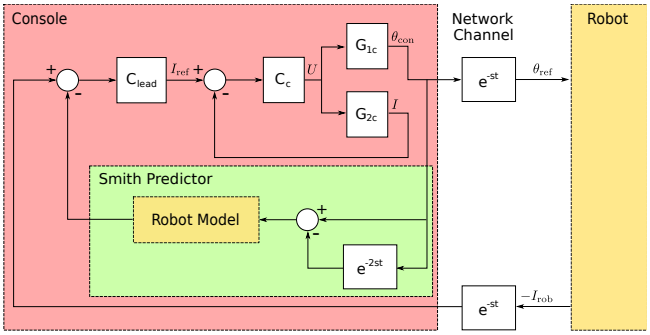


Figure 5: *Illustration of how the lead compensator and Smith predictor is incorporated into the control scheme. The lead compensator makes sure that the open loop system has a phase margin of 45° or above and the Smith predictor handles the varying delays imposed by the network channel.*

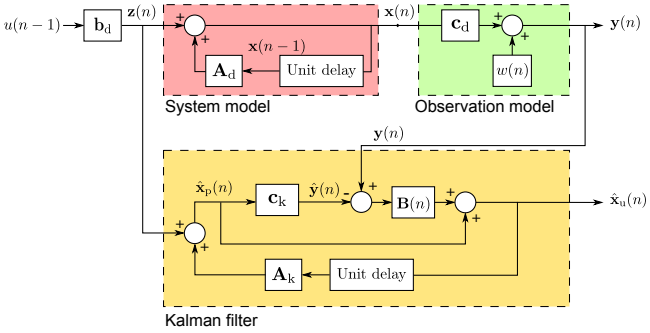


Figure 6: *Illustration of how the system and observation model are connected to the Kalman filter.*

continuous time.

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + \mathbf{b} \cdot u, \qquad \mathbf{x} = \begin{bmatrix} i \\ \omega \end{bmatrix} \tag{7}$$

$$y = \mathbf{c} \cdot \mathbf{x}, \qquad y = \omega \tag{8}$$

This state space model is used to estimate the current based on the voltage and the velocity of the motor. By adding a Kalman filter the estimation can be even more accurate. For implementation purposes the model is as shown in Eq. 9 and 10. The sampling frequency is chosen fast enough to ensure that tha dynamics of the motor is not suppressed.

$$\mathbf{x}(n) = \mathbf{A}_d \cdot \mathbf{x}(n-1) + \mathbf{b}_d \cdot u(n-1) \tag{9}$$

$$y(n) = \mathbf{c}_d \cdot \mathbf{x}(n) \tag{10}$$

The Kalman filter model is seen in Eq. 11, 12 and 13 and the design is illustrated in Figure 6, which shows how the Kalman filter uses the model to predict the states $\mathbf{x}(n)$.

$$\hat{\mathbf{x}}_p(n) = \mathbf{A}_k \cdot \hat{\mathbf{x}}_u(n-1) + \mathbf{z}(n), \qquad \hat{\mathbf{x}} = \begin{bmatrix} \hat{i} \\ \hat{\omega} \end{bmatrix} \tag{11}$$

$$\hat{y}(n) = \mathbf{c}_k \cdot \hat{\mathbf{x}}_p(n), \qquad \hat{y} = \hat{\omega} \tag{12}$$

$$\hat{\mathbf{x}}_u(n) = \hat{\mathbf{x}}_p(n) + \mathbf{B}(n) \cdot (y(n) - \hat{y}(n)) \tag{13}$$

With this filter design it is possible to use the velocity and voltage to estimate the current and thereby the torque.

C Test Description

Because of the extent of the project and the resources available it was not possible to implement all features wanted in the requirements, therefore only the following three tests of the system is conducted. Each tests is performed using a local network at Aalborg University. the delays are are generated with the use of the network simulator Netem, such that they are both known and reproducible[3]. Only data from one joint will be presented in this paper.

C.1 1st Test - Position Tracking

Position data from the motors of both surgical console and surgical robot are logged, while the test are conducted with delays of 0, 100, 200 and 400 ms. In each delay the joystick is moved 45 degrees right of the centre position, then 45 degrees left of the centre position and finally moved back into the centre position. This is done to test how well the joysticks position is matched by the robot. For each test position the grip is loosened for two seconds, to test if instability is present.

C.2 2nd Test - Torque Tracking

Estimated current at both surgical robot and surgical console is logged. The test is performed with 0, 100, 200, and 400 ms delays. The test is conducted by fixating the end-effector and then applying force to the joystick. As the surgical robots motor is now fixated, the current, and thereby torque will rise, and it will be possible to see how well the surgical console tracks and applies this.

C.3 3rd Test - Parameter Estimation

The actual motor current is measured and logged together with the estimated current. force is applied to the joystick, in one direction and then in the opposite direction. The end-effector is the fixated and the force is applied to the joystick in a similar manner. It will then be exposed how well the estimator is able to estimate the actual motor current and its direction.

III. RESULTS

The results are presented as graphs of the transient behaviour to inputs described in the previous section. Only the tests where the round trip time is 0 and 400 ms are shown. This is done as 400 ms is the maximum allowable delay in the system, and for systems with less delay, the transient behaviour is better in both theory and tests. Throughout all test the latency measurer built into the system was tested and was in all cases found functional.

A 1st Test - Position Tracking

First, a test with 0 ms round-trip time is seen in Figure 7 on the following page. It shows how the robot tracks the position of the joystick. The steady state error seen on the position tracking is most likely due to an insufficient integral term, resulting in the more aggressive controller on the joystick compensating for the error once it is released. By fine tuning the robot's controller, the tracking should be improved without having the joystick move. Increasing the round-trip time to 400 ms the response is as seen in Figure 8 on the next page. While the system is still stable, a damped oscillation is now present. The steady state behaviour appears to suffer from the same problem as without delay, that is the joystick is compensating for the slower integral term on the robot.

[3]The round-trip time between the two computers, is found to have mean 0.4 ms and standard deviation 0.1 ms. This delay is treated as zero during the tests.

B 2nd Test - Torque Tracking

When the movement of the robot is restrained and the test is performed without delay the systems' responses can be seen in Figure 9 on the following page. There is a clear overshoot of approximately 40% with no significant steady state error. Additionally, it appears that the motor input reaches saturation as the position is tracked poorly compared to the previous test while the current for the most part is bounded at ± 115 mA. Adding a round-trip time of 400 ms will result in Figure 10 on the next page. The most notable effect of the delay is again a damped oscillation, now for the current, whenever the joystick is moved. Though it is still stable.

C 3rd Test - Parameter Estimation

The parameter estimation test compares the estimated with the measured current on the robot side of the setup. The comparison is depicted in Figure 11. It shows that the current measured by use of the motor driver is only positive and cannot be below 50 mA. This is caused by the motor driver, and the way its current sensor works.

As seen, the current estimation tracks the measured current well. In the scenario where the robot is fixated (after 6 seconds) the estimation does not reach the measured current. This is assessed to be caused by simplifications in the model. Otherwise it is seen how the estimated current describes a lot of the dynamics of the measured.

IV. CONCLUSION

Throughout this paper the design and implementation of a surgical robotic system have been presented. Security measures and functionalities have been used in the design, however not all design functionalities and requirements have been implemented and thereby not tested. The implementation is done in a modular manner onto two PC's running low-latency Ubuntu kernels, whilst simulating the delay in the network by Netem.

The control design makes it possible to each individual loop separately, by ensuring internal stability is guaranteed within these. Hereafter a single path of the outer loop is examined. The outer loop had a phase margin of 5°, and a lead compensator was used to raise the phase margin above 45°. Also to decrease destabilising effects of the time delays Smith predictors are implemented. As the current was not measured on the motor drivers, a Kalman filter was implemented to estimate this, thereby a estimated torque was also acquired.

The results of this paper was obtained by testing the core functionalities of a surgical robotic system. The results impose it possible to design a control scheme that is stable when inflicted with time delays of considerable sizes. It is also shown it is possible to apply force feedback to the user of a surgical robotics system, by only using the input and position of the motor, to estimate the current.

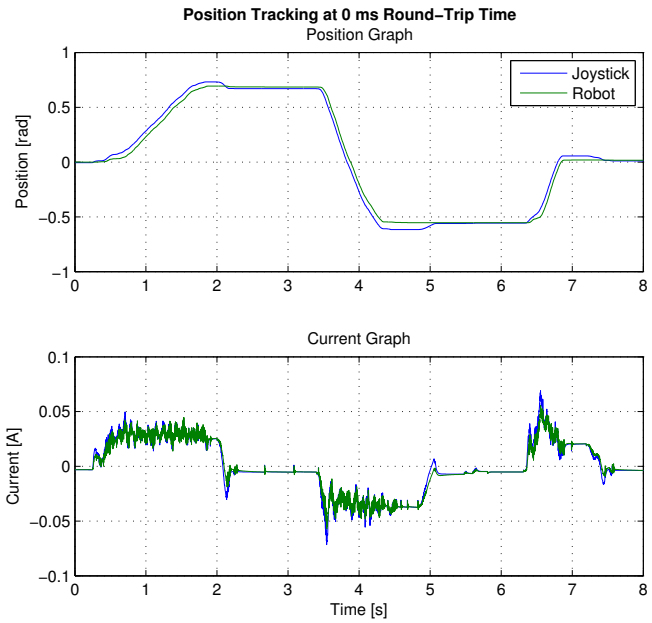


Figure 7: *Test of system at 0 ms round-trip time and robot moving freely. The first plot shows how the position of the joystick and robot behaves, while the second plot shows the estimated current on the robot and the estimated current on the joystick.*

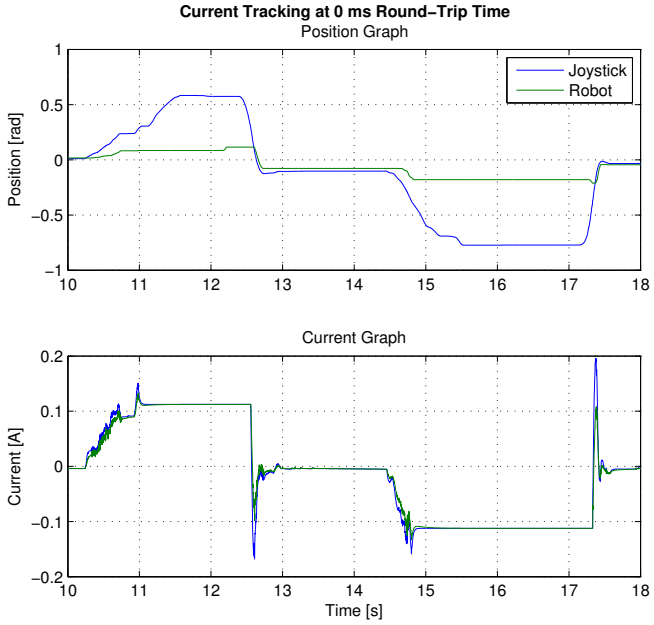


Figure 9: *Test of system at 0 ms round-trip time and robot restrained.*

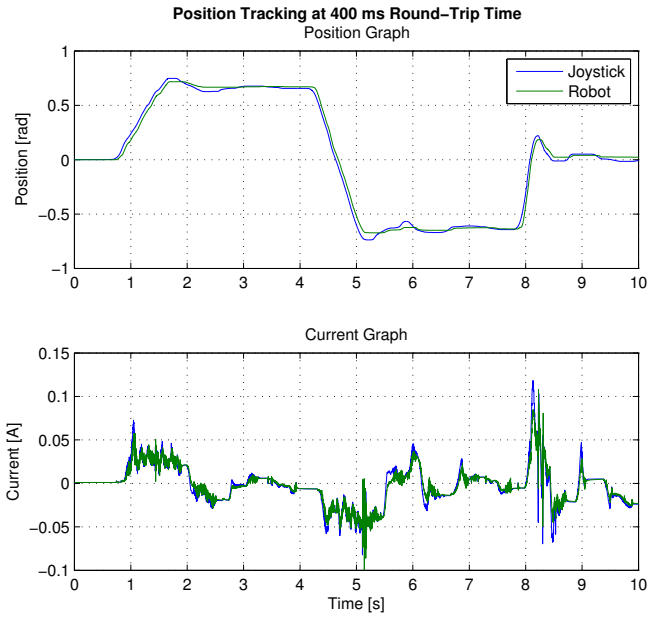


Figure 8: *Test of system at 400 ms round-trip time and robot moving freely. The time-axis for the robot is shifted by 200 ms in order to improve comparability.*

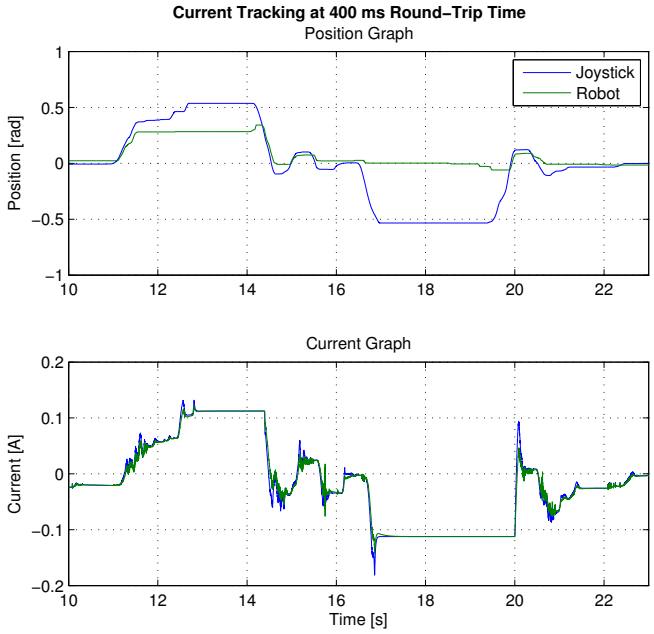


Figure 10: *Test of system at 400 ms round-trip time and robot restrained. The time-axis for the robot is shifted by 200 ms in order to improve comparability.*

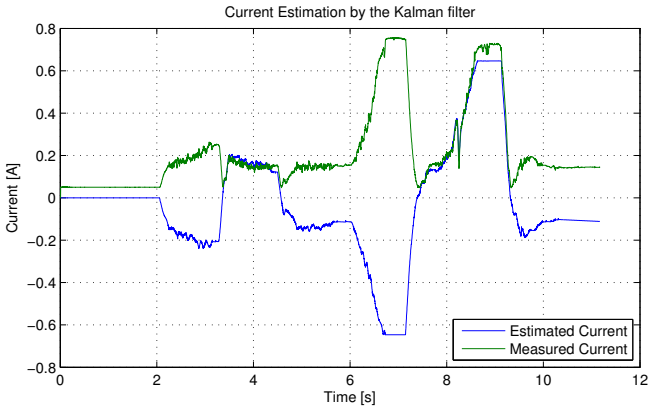


Figure 11: *Test of the parameter estimator. The joystick is moved back and forth, and from time 6 s, the robot is fixated. The current sensor can only measure in absolute values and cannot show below 50 mA.*

V. PERSPECTIVES

The position of the robot corresponds to the position of the joystick. The relationship between joystick and robot position in the test setup is 1:1. This relationship in commercial products, like the da Vinci Surgical System, is 5:1 seen from the console side, as this filters shaky hand movements and makes it possible to conduct more precise procedures. Using robotics, it is possible for the surgeon to perform under more comfortable conditions, such as sitting in a chair and avoid wearing caps, face mask and gloves. However, having to also apply the same force, instead of merely moving a joystick, could result in unnecessary fatigue. Depending on the procedure performed it could be beneficial to be able to reduce the amount of force fed back, at the surgeon's discretion, when it is irrelevant for the quality of the procedure. Taking this a step further, it could prove useful to make on-the-fly changes to feedback ratio in both directions. Meaning that the surgeon would also be able to enhance the sense of touch for when performing the most delicate procedures, incorporating the studies of [10], such that telesurgery in the future could utilise and enhance the surgeon's sense of touch instead of completely eliminating it as it currently does.

Additionally, control methods not relying on knowing the time-delay could be studied for a more robust solution to the problems of latency. Similarly, further research into the effects of packet loss, and how it can be handled, could also be relevant.

REFERENCES

[1] J. Marescaux et al., "Transcontinental robot-assisted remote telesurgery: Feasibility and potential applications," *Annals of Surgery*, vol. 235, no. 4, pp. 487–492, Apr. 2002.

[2] R. Rayman et al., "Long-distance robotic telesurgery: A feasibility study for care in remote environments," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 2, no. 3, pp. 216–224, Sep. 2006.

[3] J. Arata et al., "Impact of network time-delay and force feedback on tele-surgery," *International Journal of Computer Assisted Radiology and Surgery*, vol. 3, no. 3, pp. 371–378, Jun. 2008.

[4] J. M. Thompson, M. P. Ottensmeyer and T. B. Sheridan, "Human factors in telesurgery: Effects of time delay and asynchrony in video and control feedback with local manipulative assistance," *Telemedicine Journal*, vol. 5, no. 2, pp. 129–137, 1999.

[5] A. I. Bogani, "Low latency kernel," source: launchpad.net/~abogani/+archive/lowlatency

[6] Intuitive Surgical Inc., "da Vinci Surgical System", davincisurgery.com

[7] The Linux Foundation, "Network Emulation Tool: Netem," linuxfoundation.org/collaborate/workgroups/ networking/netem

[8] G. Beale, "Internal stability," source: teal.gmu.edu/~gbeale/ece_720/internal-stability.pdf

[9] S. Padmakumar, A. Vivek and R. Kallol, "A tutorial on dynamic simulation of DC motor and implementation of Kalman filter on a floating point DSP," *World Academy of Science, Engineering and Technology*, vol. 53, no. 126, pp. 781–786, 2009.

[10] G. De Gersem, H. Van Brussel and F. Tendick, "Reliable and enhanced stiffness perception in soft-tissue telemanipulation," *The International Journal of Robotics Research*, vol. 24, no. 10, pp. 805–822, Oct. 2005.

[11] H. Arioui, A. Kheddar and S. Mammar, "A model-based controller for interactive delayed haptic feedback virtual environments," *Journal of Intelligent and Robotic Systems*, vol. 37, no. 2, pp. 193–207, 2003.

[12] R. Rayman , K. Croome , N. Galbraith , R. McClure , R. Morady , S. Peterson , S. Smith , V. Subotic , A. Van Wynsberghe and S. Primak, "The International Journal of Medical Robotics and Computer Assisted Surgery," *Journal of Intelligent and Robotic Systems*, vol. 2, pp. 216–224, 2006.

[13] Jumpei Arata , Hiroki Takahashi , Shigen Yasunaka , Kazushi Onda , Katsuya Tanaka , Naohiko Sugita , Kazuo Tanoue , Kozo Konishi , Satoshi Ieiri , Yuichi Fujino , Yukihiro Ueda , Hideo Fujimoto , Mamoru Mitsuishi and Makoto Hashizume, "Impact of network time-delay and force feedback on tele-surgery," *International Journal of Computer Assisted Radiology and Surgery*, vol. 3, no. 3, pp. 371–378, 2008.