PREFACE

The motivation for writing the thesis comes from a curiosity in how humans and robots can be collaborative and creative in union. This curiosity was sparked during a study that set out to investigate how robots could be brought into the context of the gastronomical kitchen. The work done during that study became the foundation for the work done in the thesis, however with a different focus, human-robot collaboration in the field of architecture. Our focus throughout the thesis has been on how a human and robot can collaborate in a design process – how creativity is the result of the shared work focus of the architect and the robot. As we began to gain an understanding of the domain through observations, we were puzzled by how architects interact with the robot, a mesmerizing physical construct, which was done through CAD/CAM tools and parametric design tools. The necessity of this digital interaction platform required the architects to turn their back to the physical world and let their creativity be limited in a digital world. The purpose of this book is to give the reader an insight into our process of investigating human-robot collaboration in a creative process and insights from the experience and knowledge that we obtain during this process.

This book marks the end of four and a half months of working with researchers, architects and supervisors - exploring human-robot collaboration in a creative process. We would like to express our most sincere gratitude to those who have participated, helped and cheered us on – helping us progress through challenges that arose during the process.

Thesis supervisor
Ole Caprani

Project supervisor
Tim Merritt

External examiner
Erik Grönvall

Søren Pedersen
20105434

Christian Østergaard Laursen
20105657

ABSTRACT

The motivation for this thesis came from a previous study within the gastronomical kitchen and how robots could be active partners, supporting the chef's daily work, which sparked our curiosity for robotic agents as collaborators in creative processes. Looking at the use of robots in creative domains, the field of Architecture have embraced robots as a tool for exploration and fabrication in their design process. Current practice involves the use of parametric design tools and offline programming of the robots, which is time-consuming and creates a high entry barrier for architects.

During observations into the use of robots in the field of architecture, numerous and severe limitations was observed in the way that architects interact and use these highly complex machines in their design process. The architect's use of parametric design tools created a disconnect between the physical world and digital world, reducing the creativity of their craft to the digital world. Thus, the highly iterative process of designing new forms was limited by the existing tools available.

Using a research-through-design approach, we created several exploratory prototypes that investigated how the disconnect between the physical world and the digital world could be reduced. The exploratory prototypes were guided by an initial framework using dimensions based on identified key aspects within human-robot collaboration and human-robot interaction. These prototypes were evaluated by exploring form in granular materials.

Using the framework as an analytical tool, we see how behaviours affect the sharing of control between robot and human and how this can augment the collaboration and in turn, the creative process. Throughout the interaction between the two, the roles of the robot can shift based on the level of autonomy and how the robot intervenes or supports the architect based on a shared goal. We see that a in-air gestural interface and a tangible interface reduces the disconnect between the physical and digital world; effectively reducing the overhead between iterations within the exploration stage of the design process. The thesis serves as an introduction to human-robot collaboration in creative processes, taking the field of architecture and form exploration as the basis for the study and reveals future potentials for human-robot teams.

ACKNOWLEDGEMENTS

Firstly, we would like to thank thesis supervisor, Ole Caprani, who have been an exceptional mentor - providing guidance and extraordinairy and straight-to-the-point feedback on our process and work. We would like to extend the same gratitude to our project supervisor Tim Merritt, who have been our domain expert, source of support and inspiration throughout the process (even late-night delivery of 3D prints directly to the office desk).

We thank the staff of the Computer Science Department for providing us with a workroom and the equipment needed to work focused on our project - we, especially, appreciate the support provided by Special Consultant Søren Poulsen.

We have benefited greatly from a number of external reseachers, who have gone above and beyond helping us understand the field of Robotics in Architecture.

We would like to thank researcher Michael Knauss for devoting time to share his work and past projects, discussing the future for the field. We also extend our gratitude to researchers Johannes Braumann and Sigrid Brell-Cokcan for their Utzon(x) lecture and afterwards, taking time to talk about their experiences within the field and its' future potentials. We also thank Johannes Braumann for providing a beta-version of their KUKA|prc software.

During our visit to Tachi Lab at Keio Media Design in Tokyo, researchers Roshan Peiris and Charith Fernando took time out of their busy schedule to present their haptic teleoperation of robots projects.

During the implementation of the KUKA robot, researcher Lars Ivar Hatledal went out of his way, to ease our frustrations and overcome the challenges we faced.

In the end, we would like to thank our family and friends for supporting us during the thesis - a quick shoutout to our fellow thesis-writing comrade, Michael Ha, for keeping us company during the late nights.

TABLE OF CONTENTS

PREFACE I
ABSTRACT II
ACKNOWLEDGEMENTS III
TABLE OF CONTENTS IV

INTRODUCTION 2

ROBOTICS IN ARCHITECTURE 3
HUMANS COLLABORATING WITH ROBOTIC AGENTS IN A CREATIVE CONTEXT 3
RESEARCH PROBLEM AND QUESTIONS 3
RESEARCH APPROACH 4
CONTRIBUTION 4
STRUCTURE OF THE THESIS 4

A HISTORIC PERSPECTIVE OF ROBOTIC AGENTS 8

PRELIMINARY EMPIRICAL STUDY 16

ROBOT-SUPPORTED FOOD EXPERIENCES 17
Robotics and Gastronomy 17
Motivation 18
Exploratory Design Prototypes 18
Inspiration for the Thesis 20
ROBOT-SUPPORTED ARCHITECTURE 21
Digital Materiality and Digital Fabrication 21
Current Practice 23
Design process and Form Exploration 24
Interview Findings 26
SUMMARY OF EMPIRICAL FINDINGS 28

HUMAN-ROBOT COLLABORATIVE CREATIVITY 30

CREATIVITY 31
The Definition of Creativity 31
The Individualist Approach 31
The Sociocultural Approach 32
The Creative Process 32
HUMAN-ROBOT COLLABORATION 33
Collaboration and Robotics 34
Human-human collaboration 34
Robotics role in collaborative tasks 35
Common Ground and Joint Intention 37
Communication 38
Communication of Intention 38

Strengths of Non-Verbal Communication 39
HUMAN-ROBOT INTERACTION 40
Interfaces 41
Natural User Interface 42
Tangible User Interface 42
Communication 43
SUMMARY OF CREATIVITY, HRC AND HRI 45
ROBOTS IN ARCHITECTURE 48
SOFTWARE TOOLS FOR PROGRAMMING ROBOTS IN THE FIELD OF ARCHITECTURE 49
RELATED RESEARCH IN ARCHITECTURE AND ROBOTICS 50
Towards a Taxonomy and Framework for Architectural Robotics 50
Approaches to the use of Robotics in Architecture 51
Robots as a Tool for Digital Fabrication 53
Interactive Fabrication: New Interfaces for Digital Fabrication 53
Augmented Reality and the Fabrication of Gestural Form 55
Interlacing 55
INTERVIEWS WITH DOMAIN EXPERTS 57
Interviews with Johannes Braumann and Michael Knauss 57
RESEARCH PROBLEM AND SUMMARY 59
Research question one 59
Research question two 59
Research question three 59
EXPLORING THE DESIGN SPACE OF HUMAN-ROBOT COLLABORATION IN ARCHITECTURE 62
PRELIMINARY ROBOT PROTOTYPE 63
Technical Implementation 63
Evaluation of Interaction 64
PLATFORM FOR EXPLORATIVE PROTOTYPING OF HUMAN-ROBOT COLLABORATION 66
The KUKA KR6 R700 sixx Robot 66
Setup of Work Area 67
Connecting, Calibrating and Configuring the Robot 69
Connecting the Controller 69
Calibrating the Robot 69
Configuring Safety 70
Interfacing with the Robot 71
Challenges and Reflections on the Setup Process 73
INITIAL FRAMEWORK FOR HUMAN-ROBOT COLLABORATION IN FORM EXPLORATION IN ARCHITECTURE 74

A Preliminary Framework 74
THE DEVELOPMENT OF THE EXPLORATORY PROTOTYPES 79
Keyboard Interaction Prototype 79
Technical Implementation 79
Input-Output Mapping 80
Type of Interaction 81
Level of Autonomy 81
Role of the Robot 81
Mouse Interaction Prototype 81
Technical Implementation 82
Input-Output Mapping 82
Type of Interaction 82
Level of Autonomy 83
Role of the Robot 83
Leap Motion Interaction Prototype 83
Technical Implementation 83
Input-Output Mapping 84
Type of Interaction 85
Level of Autonomy 85
Role of the Robot 86
Tangible Blocks Interaction Prototype 86
Technical Implementation 86
Input-Output Mapping 88
Type of Interaction 89
Level of Autonomy 89
Role of the Robot 90
Summary of the Explored Prototypes and Lessons Learned 90
SAND EXPERIMENT: CURVE EXPLORATION IN A GRANULAR MATERIAL 90
The Setup 91
Keyboard Interaction Prototype 91
Mouse Interaction Prototype 92
Leap Motion Interaction Prototype 92
Tangible Blocks Interaction Prototype 94
Tangible Blocks + Leap Motion Interaction Prototype 95
SUMMARY OF CHAPTER 98
EVALUATION OF PROTYPES AND FRAMEWORK FOR ROBOTICS IN ARCHITECTURE 100
COMPARING EXISTING PRACTICE WITH PROTOTYPES 101
Existing Practice of using KUKA|prc – a parametric design tool 101
Strengths and Weaknesses of the Explored Interaction Technologies 102

Keyboard-based Interaction Prototype 102
Mouse-based Interaction Prototype 103
Leap Motion-based Interaction Prototype 104
Tangible Blocks Interaction Prototype 104
Tangible + Leap Motion Interaction prototype 105
EVALUATION OF THE INITIAL FRAMEWORK 105
Usability of the Framework 105
The Framework as a Generative Tool 105
The Framework as an Analytical Tool 106
Leap Motion Prototype 106
Tangible Blocks Prototype 107
Summary of Framework Evaluation 109
Discussion of the initial Framework 109
DISCUSSION, CONCLUSION AND FUTURE WORK 112
DISCUSSION OF EXPLORATORY DESIGN PROCESS 113
DISCUSSION OF THE COLLABORATIVE AND CREATIVE ASPECT 115
CONCLUSION OF THE THESIS 117
Research questions 117
FUTURE DIRECTIONS FOR ROBOTS IN ARCHITECTURE 118
BIBLIOGRAPHY 122
APPENDIX 138
APPENDIX 1 - INTERVIEW GUIDE FOR MASTER AND PH.D. 140
APPENDIX 2 - INTERVIEW GUIDE FOR JOHANNES BRAUMANN 141
APPENDIX 3 - INTERVIEW GUIDE FOR MICHAEL KNAUSS 142


TMO 150
3HAC032140-001

1 Introduction

In this first chapter, we introduce and outline the domain of the thesis. Secondly, we present the identified overall problem supported by the three research questions that shape our work. Next, the adopted approach for exploring and solving the problem in question will be elaborated on, along with the proposed contribution of the thesis. Lastly, we will present and outline how we have structured the thesis.

ROBOTICS IN ARCHITECTURE

Currently, industrial robots are being used for various methods of digital fabrication by architects. However, the process of using these robots are rather complex and tedious for both the expert and novice architect. Industrial robots have been used for very specific purposes with a focus on the efficiency and repetitive nature of it. However, as the robots are emerging as a common technology throughout the world of architecture, the scope of use is also changing as the diversity of use scenarios become greater. Thus, robots are now being used early in the creative process for form exploration.

However, even though the scope of the robot use is changing, the way the architect program and interact with it has not. New tools have been developed, but they are still merely parametric design tools that requires an intermediate level of knowledge in the use of software tools, such as Grasshopper and KUKA|prc in order to sketch out a design (Grasshopper, 2015) (HAL Robotics, 2015) (Brell-Cokcan & Braumann, 2010) (Brell-Cokcan & Braumann, 2011). Our work set out to explore various methods of interacting with robots in the creative design process of form exploration and how the robot can contribute to the creative process working as a peer in relation to the architect.

HUMANS COLLABORATING WITH ROBOTIC AGENTS IN A CREATIVE CONTEXT

In the current practice, architects pre-program robots and their movements, resulting in a very time-consuming iterative design process. Software tools have been introduced to lower the entry barrier for novice architects, but have only improved the process slightly. Using the current problematic practice and findings from an earlier study of robots as partners in a gastronomical setting (Laursen, et al., 2015), we look towards a new relationship between robotic agent and human, where the two share a common goal and through interaction, progress towards this goal in a creative process. We envision the two partners enter a continuous dialogue through the use of manipulative and communicative gestures in the form exploration stage of a design process; shaping the resulting creative product.

RESEARCH PROBLEM AND QUESTIONS

Motivated by the limitations of current practise within the field of robots in architecture, we set out to explore how the interaction between human and robot in the context of architecture, can be optimized regarding form exploration. In addition, we aimed to investigate how collaboration could be shaped through the manipulation of materials and objects in a shared workspace.

We now present the research questions that emerged through our preliminary work, and afterwards have guided our work exploring human-robot collaboration in a creative process, using the field of architecture as the domain of study:

- ***How can we design the interaction between human and robot, with the objective of improving the workflow of the architect's creative process?***
- ***How can the disconnect between the physical and digital world be reduced, when exploring form in granular materials?***
- ***What roles can a robot take in the***

activity of form exploration and how do these behaviours affect the architect?

To address the problem and the questions we look to explore directions through the development of prototypes. We will now elaborate on the process of the thesis.

RESEARCH APPROACH

New approaches for interaction in digital fabrication are emerging with a focus on the relationship between the digital input and physical output. However, the qualities and methods have not been fully explored. Furthermore, robots are seen as tools used in the final stages of the design process and not envisioned as entities acting on its' own with the capabilities to provide valuable input in the creative process. By adapting a research through design approach originally presented by Zimmerman et al. (2007) we have investigated the possibility of incorporating robotic agents more deeply in the creative process along with how the architect's process can be more efficient by changing the interaction to be more meaningful. This approach allows for engaging research with so called "wicked problems" that normally cannot be easily addressed through science and engineering methods (Zimmerman, et al., 2007). Thus, it helps transferring knowledge from research to practice increasing the chance that the produced knowledge will move into future research and products.

CONTRIBUTION

The contributions of the thesis are in the form of our exploratory process, describing underlying implications of designing interaction for collaboration between humans and robotic agents. Through the identification of key concepts applicable for the field of robots in architecture, we contribute a conceptual framework for guiding future work in the field and as an analytical tool for explaining human-robot teams in creative processes in architecture. We finalize our contribution by presenting future potentials for human-robot collaboration in creative processes, especially focusing on form exploration in the field of architecture.

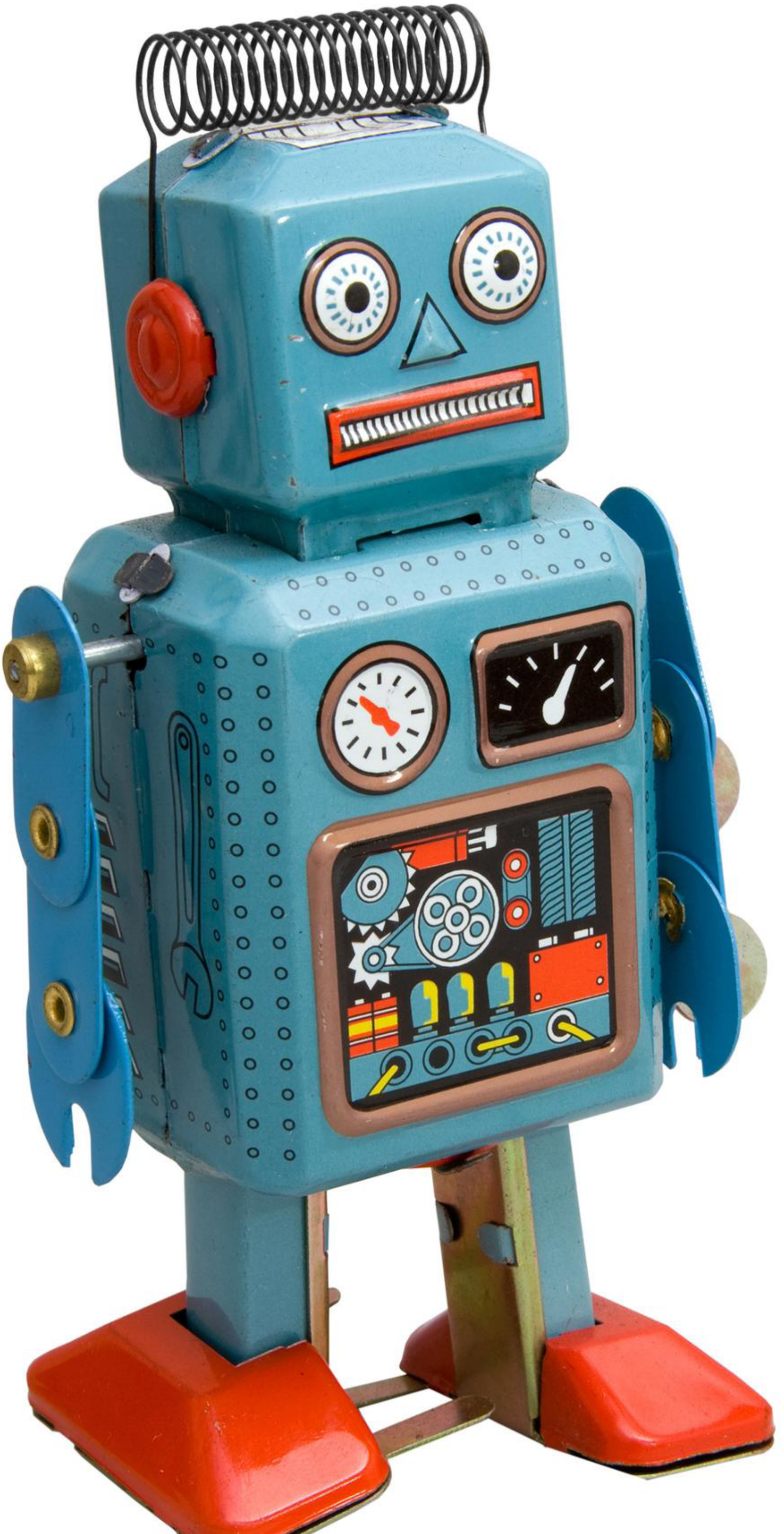
STRUCTURE OF THE THESIS

In this chapter, we have introduced the domain and the foundation for our thesis. In Chapter 2, we present the motivation for the thesis and introduce the preliminary empirical work that sketch the problem space that we have explored through observations and interviews with two architect students. In Chapter 3, we present a historical perspective of robots in order to introduce the terminology that we use throughout the thesis. Next, in Chapter 4, we present the theoretical grounding from the fields of creativity, human-robot collaboration and human-robot interaction – we identify key concepts that contribute to the understanding of the design space, which we explore.

Following this, in Chapter 5, we present related research within the field of robots in architecture and expand on this with two interviews with leading researchers from the field. Chapter 6 is divided in four larger sections; first introducing a preliminary robot prototype used for evaluating our initial assumptions, afterwards we describe the building and configuring of our experimenting platform using a desktop-sized industrial robot. Thirdly, we use the identified key concepts from Chapter 4 to sketch out an initial, conceptual framework that guides our exploratory work in the last section of the chapter.

Using the lessons learned and our newly

expanded understanding of how architects can collaborate with robots, we summarize the explorations in Chapter 7, using the framework as an analytical tool and compare to an existing software tool. We end the chapter by discussing the framework used. In Chapter 8, we summarize the thesis by discussing our process of exploring the design space, which is followed by a discussion of the collaborative and creative aspect of our explorations. We conclude on what we have learned about the design space and look at future directions and potentials.



2

A Historic Perspective of Robotic Agents

In the following chapter, we present a historical account of robotic agents, from early automations to the industrial and consumer-oriented robots of today. We want to emphasize that this is not a comprehensive review, but it serves as an introduction of the terminology regarding robotic technologies that we use throughout the thesis.

A historical account of robots would not be complete without an introduction to the Czechoslovakian word robota, meaning forced labour (Oxford Dictionaries, 2015). It was used to describe the artificial people in Karel Čapek's science fiction play entitled Rossum's Universal Robots from 1920. However, earlier examples exist of people envisioning mechanical creatures or human-like automations, such as Leonardo Da Vinci's mechanical knight, which have been thoroughly investigated and evaluated for feasability in modern times (Rosheim, 2006).

Some of the earliest examples of automata – a self-operating machine that can perform a series of predefined operations (Automaton, 2015), includes French inventor Jacques de Vaucanson's Digesting Duck from 1739 (Riskin, 2003). This stationary automaton gave the appearance of being able to consume, metabolize and defecate.

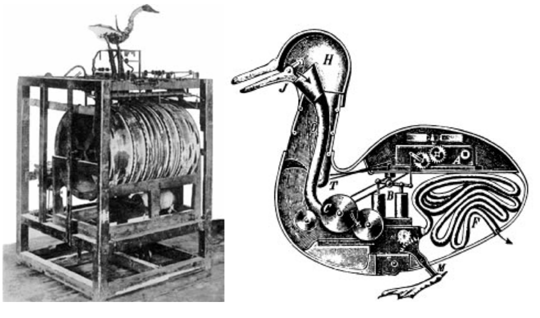


Figure 1: *Duck of Vaucanson - The Physical Construction is seen to the left. On the right, an unknown inventors imagined version of a mechanical digesting duck (Wikipedia, 2015 - Digesting Duck)*

The duck utilized a container for collecting the consumed food, afterwhich it defecated using another container, storing feces, thus only giving the appearance of digestion – the viewers' perception of the automata was of primary concern.

These automata required a human operator present in order to function, and thus, it becomes useful to distinguish the interaction between robotic agents/automata and human into two general categories, proximate interaction and remote interaction (Goodrich & Schultz, 2007). Proximate interaction is when robots and humans are co-located and remote interaction is when robots and humans are separated spatially and/or temporally.

Exemplifying these general categories, we see an early example of remote interaction, Nicola Tesla's radio-controlled boat (Tesla, 1898) – a vessel that was remotely operated by a human and that had no autonomy, i.e. without self-sufficiency or mechanics to respond to its' surroundings on its' own. Therefore, it does not fall under the modern definition of the term robot, which according to the Oxford Dictionary is:

"A machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer" (Oxford Dictionaries, 2015).

However, Tesla envisioned a more complicated, intelligent and autonomus future for automata, such as his creation, that did not continually need commands from a remote-operator. In June, 1900 Tesla published an article in The Century Illustrated Monthly Magazine, where he laid out his vision for the future of automata (Tesla, 1900). He envisioned that future automata would have a "mind of their own", which means that they will become autonomous by utilizing an internal logic for acting and reacting on external stimuli that is sensed by "sensitive organs". This vision correlates very well with the evolution of robots and this terminology is fundamental for describing robots, how they react to their surroundings and change their surroundings accordingly.

One of the earliest examples to follow Tesla's vision is Shakey the Robot (SRI International, 1984) – a project conceived by the Artificial Intelligence Center at Stanford Research Institute which began in 1966. Shakey used sensors such as sonic range finders, camera and bump detectors to sense and model its' surroundings, which was processed by on-board processors. This allowed Shakey to navigate an indoor block world using two stepper motors as drive wheels, i.e. actuators, and a third caster wheel. By responding to external stimuli, i.e. sensing that Shakey had bumped into an obstacle, and by using this information, it could change position, using actuators.

Moving away from the relatively safe, predictable and controllable indoor environment to interplanetary exploration in modern times. The introduction of behaviour-based robots allowed robots to achieve a higher level of autonomy by utilizing distributed stimulus/response pair, i.e. a behaviour, for reacting to the environment (Arkin, 1998). The use of behaviour-based architectures also had the benefit of being robust to changes in the surrounding environment, as the combination of behaviours gave robots the possibility of prioritizing a set of reactions to environment. The advancements in new software architectures and improved hardware, such as sensors and actuators has helped us explore planets, such as Mars, with planetary rovers. NASA's Mars Exploration Rover (MER) mission sent two rovers to the surface of Mars in 2004 (P. Chris Leger, 2005) – the two rovers, Spirit and Opportunity, were tasked to explore the surface concurrently for three months.

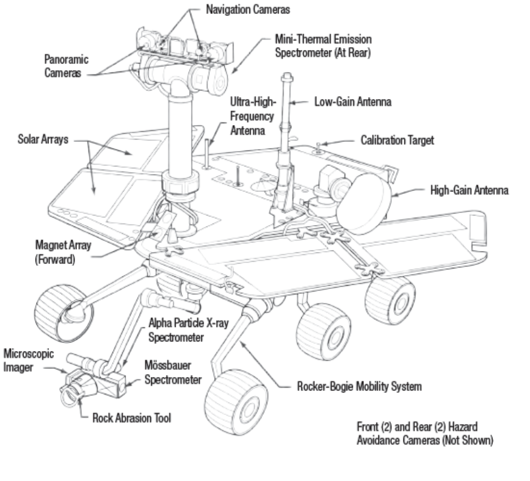


Figure 2: *Diagram of the MER rovers' sensors and actuators (NASA, 2013)*

Spirit, a 176kg, 1.6m-long, six-wheeled rover that was designed for mobility, was the first of the two to land on the surface of Mars. Spirit was equipped with a manipulator called the Instrument Deployment Device (IDD), four sets of stereoscopic cameras and a geological tool for examining the surface of Mars, called a Rock Abrasion Tool (RAT). Figure 2 shows the different sensors and actuators of the MER rover design, implementing both hazard cameras and navigational cameras for mobility purposes. As Mars can be anywhere between 78 million to 377 million kilometres from Earth, it can take between 4 and 21 minutes to retrieve or send data (NASA, 2015), this limits the ability for operators to remotely control the rovers in real-time. This limitation is overcome by pre-programming a sequence of commands and having underlying, autonomous behaviours, such as automatic hazard avoidance.

The overall purpose of the MER rovers was to explore the surface of Mars, however when considering what behaviours such robot might implement, its' primary goal, as with other living beings, is to survive. For the rovers to survive the Martian environment, they have to maintain safe, balanced energy and thermal levels prioritised above all other

***Figure 3:** Robotnaut 2 onboard the International Space Station. Teleoperators can interact, through the Robonaut, with equipment and onboard astronauts or kosmonauts. (NASA, 2013)*

behaviours, such as communication (Neilson, 2005). As opposed to the MER rovers, which were built for mobility, exploration and conducting geological experiments, NASA's Robonaut was an anthropomorphically designed robot, i.e. a humanoid robot that looks or acts like a human, built for dexterity and collaboration with astronauts (Ambrose, et al., 2000) – see Figure 3.

The Robonaut was designed as an upper body with arms and five-fingered hands, seen on Figure 3, that could wield extra-vehicular activity (EVA) tools, geological tools and medical instruments (Lovchik & Diftler, 1999). The Robonaut was created out of a need for a system that could provide human-like capabilities, operating alongside humans in the extreme environment of space, serving as an assistant on space walks, or even instead of humans. Robonaut's control system has a teleoperator interface that allow an operator from earth to control its' manipulators and tools. By utilizing feedback devices, the operator receives natural cues for sense of force and contact. The interface also allowed for maximum situational awareness by using a virtual-reality headset that was connected to the cameras mounted in the head of Robonaut. A second iteration of the Robonaut is currently undergoing testing at the International Space Station (ISS).

The Robonaut's purpose was to be versatile and dexterous, as an assistant or substitute for astronauts, however at the opposite end of the spectrum and at the core of modern assembly line automation, we find industrial robots that execute the same sequence of movements repeatedly with high speed and precision. According to the International Federation of Robotics (IFR), the worldwide demand of industrial robots is increasing with the biggest consumer of this type of robot being the automotive and electrical/ electronics industry (International Federation of Robotics, 2015). Industrial robots can be viewed as programmable manipulators that can do a great deal of different tasks, which

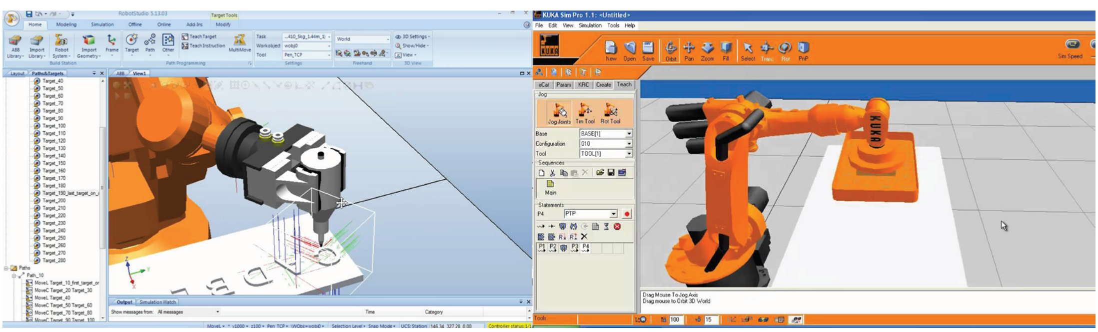


***Figure 4:** On the left, a screenshot of the RobotStudio software (Pathlist on the left, 3D viewport on the right). On the right, a screenshot of the KUKA SimPro software (Move command creator on the left, 3D viewport on the right) (Youtube, 2011 - KUKA) (Youtube, 2011 - RobotStudio)*

was done by manual labour before, through a pre-taught sequence of movements and actions (Ayres & Miller, 1981). These tasks include packing, sorting, spot welding, spray painting or cutting, benefitting of qualities such as predictability, reliability, precision and relative resistance to hostile environments - reducing manufacturing costs caused by reducing manual labour. Due to the repetitive nature of industrial robots, the work-sequence is static after being pre-programmed, i.e. offline programmed, done through advanced software such as ABB RobotStudio (ABB Robotics, 2015), KUKA WorkVisual/SimPro (KUKA, 2015) or Universal Robots URSim (Universal Robots, 2015) – see Figure 4. These software tools often include computer-aided manufacturing (CAM) features, allowing programmers to run code on a virtual robot in a 3D world and checking for collisions and reachability - the goal being a quick cycle, i.e. the time it takes for a robot to complete a full sequence of movements for one task.

The size, capacity and mechanical structure of the robots vary, depending on the specific task needed to be completed, these are classified as follows: articulated robot, SCARA robot, Cartesian robot and parallel robot (International Federation of Robotics, 2015) as seen on Figure 5. Robots can also be seen in the healthcare industry, e.g. the surgical theatre, allowing surgeons to operate with high precision by translating their hands' movement into very small movements of a robotic arm. An example of this is the da Vinci Surgical Robot (Wikipedia, 2015), a common use for this robot, is radical prostatectomy, removing the prostate gland and surrounding tissue, i.e. curing prostate cancer. Robots, however, have also moved from industry and research into the consumer's home as well, either as service robots (also called household, domestic or personal robots) for household chores, such as vacuuming or therapeutic/social robots for elderly or people with disabilities.

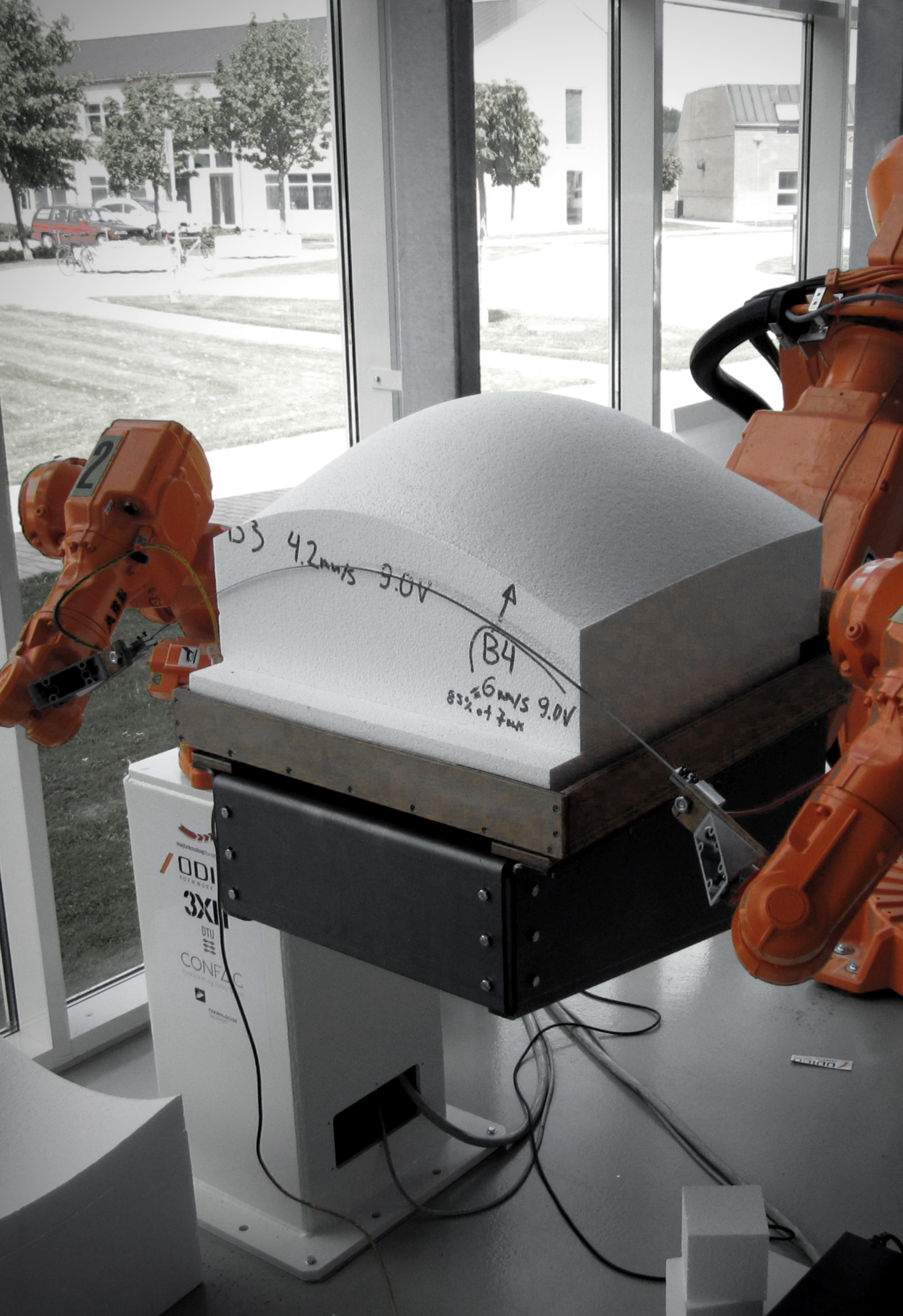
Examples of service robots includes the iRobot Roomba (iRobot, 2015), the Dyson 360 Eye (Dyson, 2015) or the Droplet (Droplet, 2014). These types of robots aim at automating otherwise dull tasks, such as vacuuming or watering plants – e.g. using data to schedule watering and analyse what parts of the garden uses most water (Droplet, 2014). These types of robots are generally autonomous and only require the user to empty their collection-bin when full, others, such as Husqvarna's Automower (Husqvarna, 2015), might need for an initial workspace setup using boundary wires.

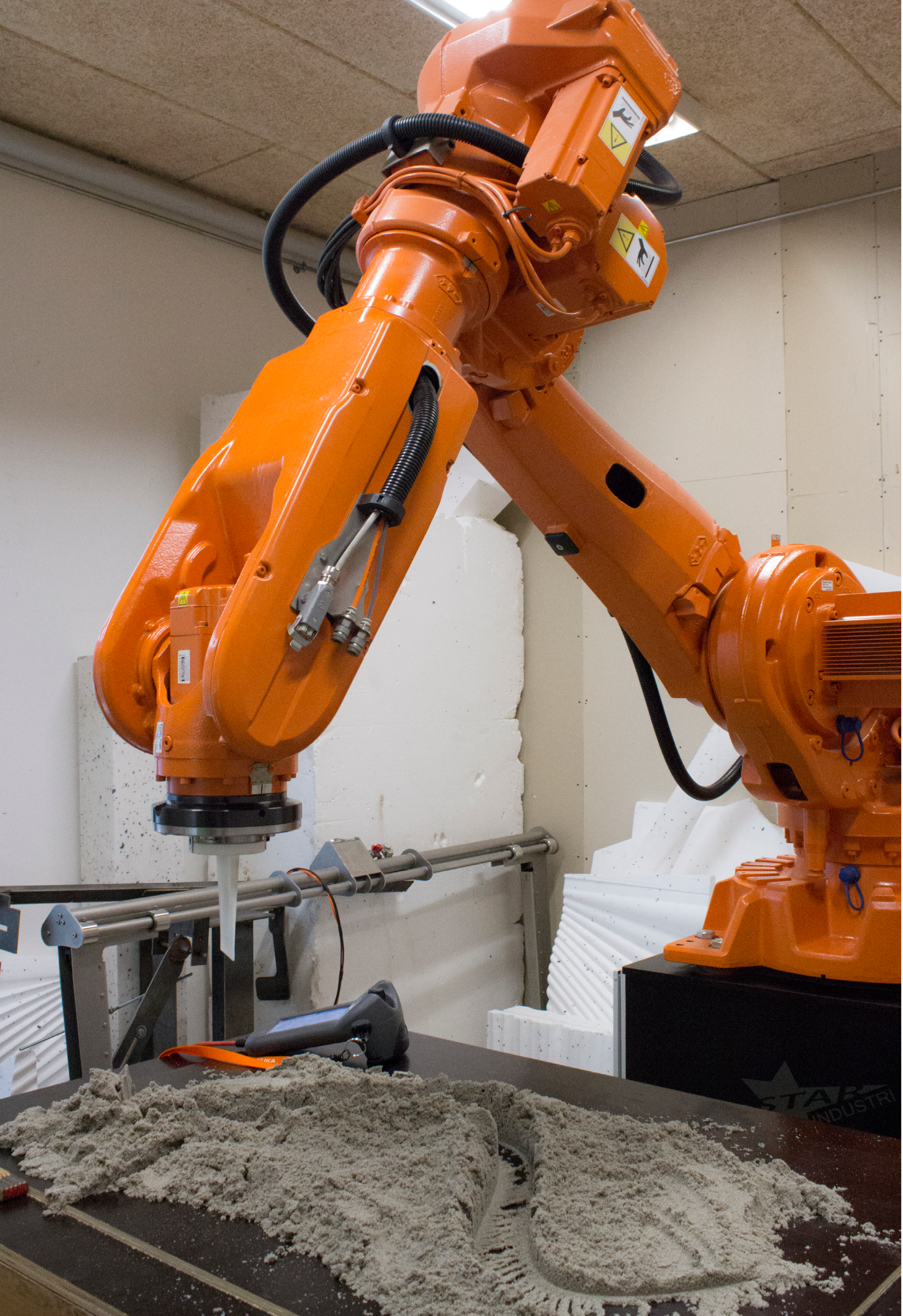
Examples of therapeutic robots include the robotic harp seal PARO (Shibata, et

Figure 5: *Types of Robots - From left to right: SCARA, Articulated, Parallel and Cartesian Robot (KUKA Robotics, 2006 - SCARA) (KUKA Robotics, 2015 - Quantec) (ABB, 2013) (Toshiba Machine Company, 2015).*

al., 2001), Popchilla (Interbots, 2012) and KASPAR (Dautenhahn, et al., 2009) for autistic children – the domain of social, assistive robots constitute a whole research field of it's own and these types of robots are being used extensively in research for evaluation of their effectiveness and helpfulness of providing physical, mental or social support (Sabanovic, et al., 2013) (Wada & Shibata, 2007) (Kidd, et al., 2006) (Wada, et al., 2005) (Wada, et al., 2004).

In summary, we see that robots have taken on various roles and using Scholtz's taxonomy of robot roles (Scholtz, 2002), we see how robots take on roles throughout history, such as a surgical robot being a supervisor during surgery or the Robonaut as a peer of the astronauts. Scholtz's taxonomy will be described briefly in Chapter 4. These roles, along with the design of the robots, have influenced how humans perceive and interact with robots – from the highly autonomous iRoomba cleaning robot to the tele-operated Robonaut that can be perceived as a peer when assisting astronauts.


4.2mm/s 9.0V
B4
≈6mm/s 9.0V
3XN
DTU
CONFAC



3

Preliminary Empirical Study

The following chapter will be divided into two sections consisting of the study of robotics in gastronomy and an empirical study into praxis of robotics in architecture. We will present themes, motivation, method, and outcome of the study and how we can use the insight gained for the thesis as a preliminary study in the first section. The second section will summarize and present the preliminary observations and interviews with architect students who have intermediate experience in using robotics for form exploration.

ROBOT-SUPPORTED FOOD EXPERIENCES

In the following section we will summarize the study of robotics in gastronomy and the motivation behind the study. Next, the adapted research approach will be described along with the developed prototypes and how the lessons learned contributes to this thesis and future work.

Robotics and Gastronomy

The study began as an exploratory study with the goal of testing the newly acquired ABB IRB120 (ABB, 2015) industrial robotic arms at Aarhus School of Architecture - See Figure 1 and 2. As robotics have mostly been seen as a tool for the industry – especially the automotive industry – we saw an opportunity to incorporate them into more untraditional domains challenging the perception of the use of robotic agents as an everyday tool or assistant in highly complex human environments.

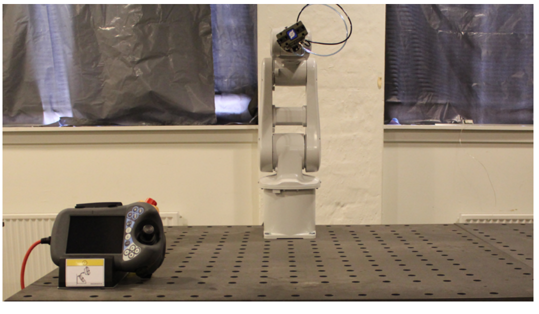


Figure 1: *The industrial robot and the basic setup of the working table along with the control pendant.*

The authors idea was to explore how robotic agents could enhance or challenge the creative process of chefs in the modern gastronomical kitchen, which is much inline with the scope of the thesis. Which is to explore how robotic agents can be incorporated more actively in the creative process of architects. Whereas much focus in research and in the industry, in general, has been on replacing the human worker, to bring in the benefits of robots, specifically efficiency and speed for repetitive tasks; our goal was not to solve current problems linked to these aspects. However, we sought to explore and imagine how robotic agents could support aesthetic and pleasurable experiences with food, and support the creative process in the kitchen (Laursen, et al., 2015). However, it is only the latter aspect we intend to focus on in the thesis.

Figure 2: *An example of a setup for one of the exploratory prototype experiments*

The domain of robotics and food experience could have been rigorously reviewed and explored in multiple directions, however, and as mentioned above, the goal was

"[...] to engage in a design exploration to find points of divergence and possibilities for future inquiry." (Laursen, et al., 2015)

Based on this, nine exploratory design prototypes, involving preparation and serving of food with a robotic arm was developed - See Figure 3. The design prototypes took departure in themes concerning haute cuisine, "plating", and arts. Furthermore, a goal was to show some of the capabilities of robots and to trigger the imagination of chefs for possible future use of robotic agents in the modern kitchen.

We will now present the motivation for the study, which is based on a large number of related commercial and research work.

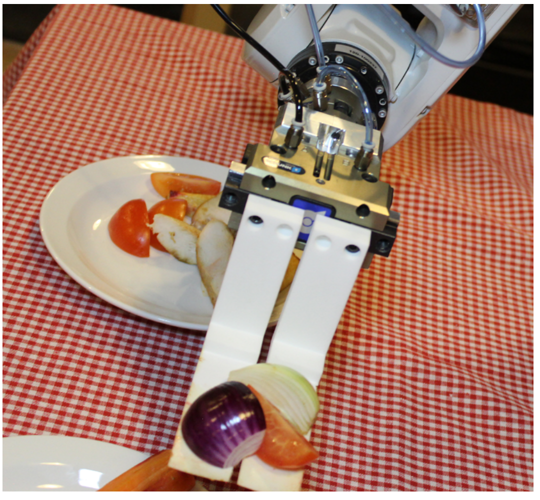


Figure 3: *The above image depicts a prototype setup. The robot assumes the role of both a chef and a server, choosing the composition of the dish at the diners' table.*

Motivation

As part of the recent movement of robots finding new roles as visible actors alongside people across society, from supporting surgeons in the mission critical surgical theatre, to providing comfort as virtual companions (Laursen, et al., 2015), we set out to explore how robotic agents could support aesthetic and pleasurable experiences with food. In addition, we sought to shed light on how robotic agents could support the chefs' creative process. Whereas much research has been focusing on replacing the human worker in many contexts, e.g. to bring the benefits of speed and efficiency from robots, our goal was not to replace the human worker or optimize existing processes but instead to investigate and explore the possibilities of having robotic agents supporting the preparation, serving and consumption of food.

In present, the food industry has been concerned that the increased usage of technology and standardization, can have negative effects, such as deskilling chefs, staff reduction, reduced labour mobility, and job losses (Laursen, et al., 2015). This served as motivational grounding for conducting the study, thus a goal to change the prejudiced assumptions about robotics and how they are viewed as entities constructed for replacing the human worker to bring speed and efficiency to the table, instead of being active partners in a creative, collaborative process along with chefs. Additionally, authors sought to engage chefs in a participatory design process, which aims at involving specialized users and develop solutions that would fit into existing practices and uncover unmet needs. This was stated as authors felt it was necessary to reach out to chefs and serving staff to examine how one might co-design experiences for the future kitchen with robotic agents as collaborators and not as a way to accelerate a workforce reduction.

Exploratory Design Prototypes

Through a research-through-design approach, authors explored the design space of robot-supported food experiences. First phase consisted of immersing oneself into the context of the kitchen through observations and interviews with chefs and serving staff. Next was the process of developing experience-prototypes (Buchenau & Suri, 2000). The purpose of this was to emphasize the experiential aspect of whatever representations are necessary to convey or live an experience related to the gastronomical kitchen. Thus, authors developed nine experience prototypes to explore, understand and communicate what it might be like to use a robotic agent as a partner, in a complex human environment such as the modern gastronomical kitchen. Further inspiration was taken from co-design workshops, where researchers present, e.g. robotic technologies to farmers so they would develop a deeper understanding and insight into the possibilities and capabilities of robots (DiSalvo, et al., 2010) (Laursen, et al., 2015). This served as a way to challenge the prejudiced assumptions that several chefs and people working with craftsmanship have, with robotic technologies, essentially a fear of being replaced or rendered useless. It would, in addition, allow the chefs and serving staff to highlight problems and sketch out possible future directions for the design of robotic technologies and experiences for the kitchen based on the experiential prototypes, which authors presented for them.

Turning to a few examples of the exploratory prototypes, we will exemplify and present some grounding for the later insights gained and contributing aspects for the thesis. Out of the nine prototypes authors developed, we will present three, two of them based on the same thematic, but with two different aspects of dimension of control (see Figure 5). The following three examples are called; Plating 1, Plating 2, and Food Visuals (see Figure 4).

The examples of Plating 1 and Plating 2 was seen as a way of delegating control between the chef and the robot. Whereas the chef is normally in full control of how the dish is being plated, authors experimented with shifting the control between the human and robot, by determining who is setting the boundaries of the plating. In the first example, it was the chef who dictated the boundaries which the robot had to work within. In the second example, the control was shifted and the robot controlled the boundaries and overall design of how the dish should be plated and thus, it was the chef who had to stay within the set of boundaries. The former example challenged the way chefs normally work, where they are in full control of the final outcome of the dish. In the latter example, the interactions of the robot forced the chef to be creative in ways that he could not fully control himself.

When composing a dish in the kitchen, the chef has to consider the fusion of the ingredients' taste, texture and colour. The combination of these defines a cultural dish, like e.g. the overall yellow colour palette of an Indian curry dish. The third example we are presenting here, authors investigated how the robot benefits of precision and repeatability of actions could create intricate mixes and shapes of colours in food. Thus, the robot would act as a tool for ideation for composition of dishes with varying colours, shapes and patterns. All this, by utilizing the precision and randomization of the robot. This randomization would exist

***Figure 4:** On the left, the chef sets the boundaries for the robot through cocoa powder. In the middle, the robot sets the boundaries for the chef to place icecream. On the right, the robot changes toolpath (moving in a shape) each time the chef adds color to the mix.*

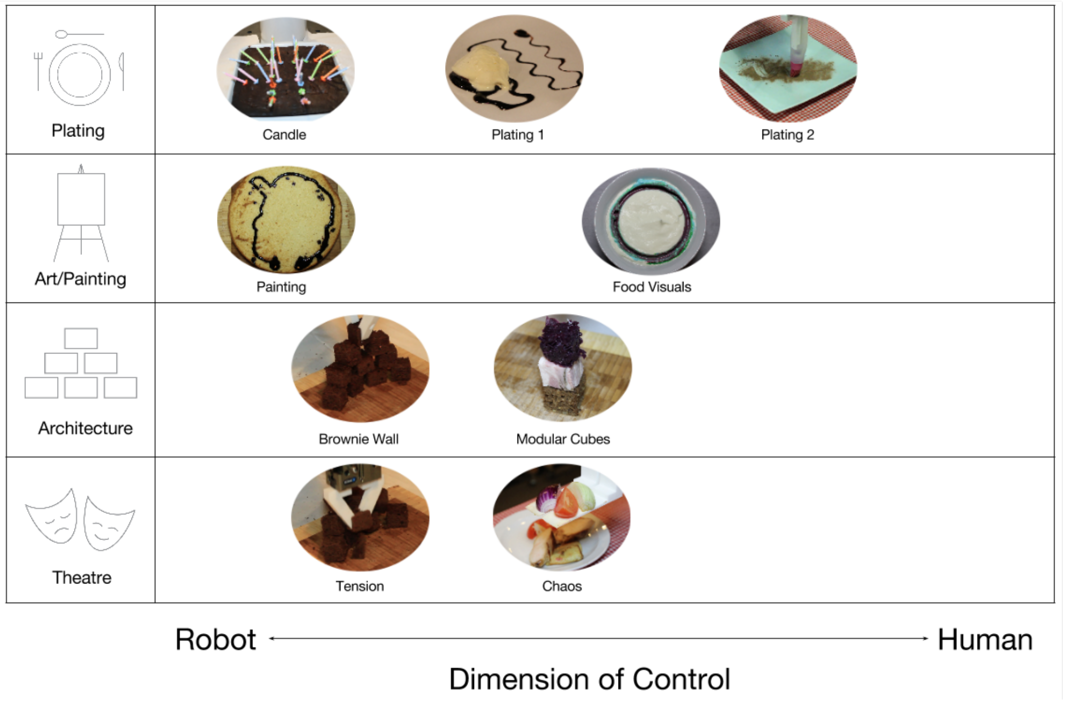


Figure 5: *Matrix of the dimension of control from Laursen et al. (2015) depicting all nine exploratory prototypes and how they relate to the dimension of control between human and robot. Only three of them are presented in this chapter.*

in pre-programmed patterns and shapes of movement. This would allow the chef to control some of the parameters while the robot would control the remaining. This would essentially form a collaboration between human and robot where control is negotiated in the beginning.

Inspiration for the Thesis

The thesis takes inspiration and motivational foundation in the previous work of robot-supported food experiences. Through this work, we find inspirations in how robotics can contribute to the creative domain of craftsmanship. In this regard, it is related to the work of chefs, and recently, in the work of architects and their design process of form exploration. From the work of the presented study, we take particular note of two primary concerns regarding the design space of robot-supported food experiences; Issues of control between human and robot, and perception of robot behaviour. Looking back at the three examples presented in the section above, we saw a shifting degree of control in the interaction between chef and robot. In the presented context of robotics in the gastronomical world, we saw a lack of common ground between chef and robot which in turn reduced the possibility of negotiation. This in turn, forced the chef to rely on the robot as an active partner. Thus, the chef has to function as a resource of information that provides the necessary information and processing in order for collaboration to succeed (Laursen, et al., 2015).

By the language of action, we saw in the example of Plating 1 (see Figure 4) that the chef created the boundaries for the robot to work within, hence the notion from above of the chef functioning as a resource providing information for the robot. This particular situation is turned around in the second example. Here we saw how the robot provided the information and the chef with the boundaries to work within. The last example showed how the chef and robot are in possession of individual aspects of the control at the same time. Thus, the chef acts on adding colour while the robot reacts by changing the pattern of movements accordingly.

Regarding the perception of robot behaviour, we noticed how people perceived the robot depending on various factors such as speed, precision and sound. The behaviour of the robot could be perceived as aggressive with the rapid movements, thus resembling someone who did not care about the task at hand. This informs the initial design space of robotics in architecture in such as way that the design of the robotic movements does not scare off novice users of robotics. The robotic agent, as both a tool and a collaborator of architectural design, should be reliable and trustworthy regarding basic safety issues. Thus, if the robotic movements are too fast and careless, it can hold back users from using it for safety reasons regarding oneself or for the robot.

ROBOT-SUPPORTED ARCHITECTURE

In the following sections we elaborate, and discuss, topics of interest obtained during sessions of observing experienced architects working with robotics in addition to semi-structured interviews. The participants were a Masters Student and a Ph.D. student in architecture from Aarhus School of Architecture. The study was focused on form exploration using fast-setting concrete.

Before elaborating the term form exploration in architectural design, we find it necessary to describe and elaborate the concepts and terms of digital fabrication and materialization as they are what constitute the use of robotics in architecture. Thus, they show how robotic agents are being used by architects in their work to explore the ever-changing possibilities of computer generated architecture (by the use of CAD programs). In the following, we will briefly introduce some of the terms used throughout the thesis in regards to architecture and robotics. The two key concepts here are; Digital Materiality and Digital Fabrication. These two concepts are interwoven and cannot be accounted for without considering the other.

Digital Materiality and Digital Fabrication

The term, digital materiality is used to describe the emerging trend of transformation in the expression of architecture (Gramazio & Kohler, 2015). Physical materiality is increasingly

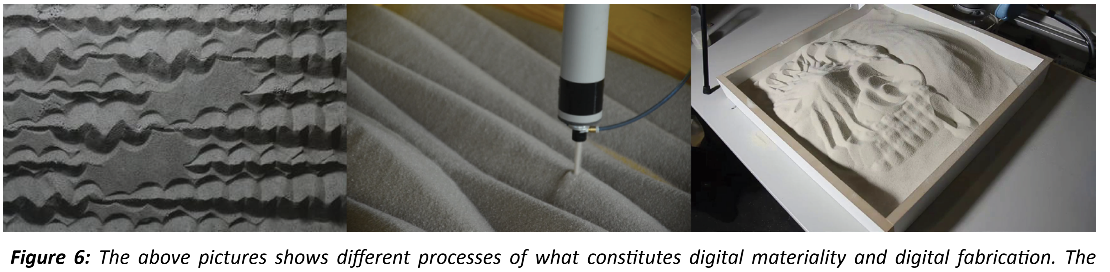


Figure 6: *The above pictures shows different processes of what constitutes digital materiality and digital fabrication. The properties of the materials are used as a fabrication constraint in the digital process. Thus, the shape of the sand is used as a mold for concrete elements.*

***Figure 7:** The finalized pre-fabricated concrete wall blocks that was created through intricate molds based on shapes drawn in sand (Gramazio Kohler Research, 2011 - Procedural Landscapes 2)*

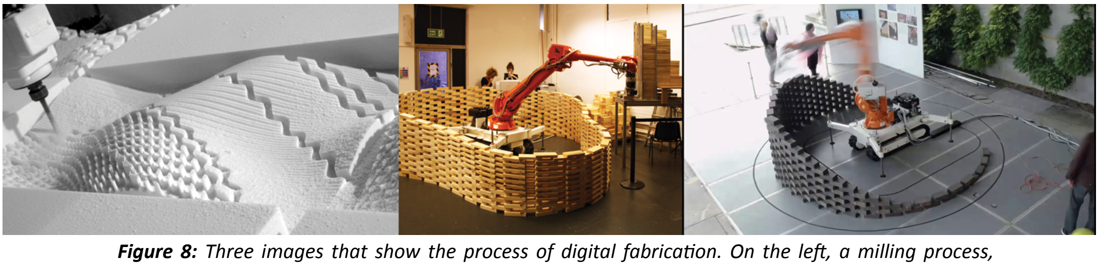


***Figure 8:** Three images that show the process of digital fabrication. On the left, a milling process, middle and right show examples of brick building (AGATA KYCIA, 2011) (Gramazio Kohler Research, 2011 - The Endless Wall) (Gramazio Kohler Research, 2012 - Stratifications)*

***Figure 9:** Snippets of the Gantenbein Vineyard Facade from the outside (left) and inside (right). Depending on the viewer's distance to the facade and the position of the Sun, one experiences the changing appearances of the facade, experiencing a fictitious glimpse inside the building (Gramazio Kohler Research, 2006)*

being enriched by digital characteristics. The interplay between the two distinct worlds of digital and material processes are evolving digital materiality. It is the synthesis between these two worlds that generates a new self-evident reality. Thus, data and material, programming and construction are interwoven according to Gramazio and Kohler (2008). Digital materiality stems from the two concepts of "digital" and "materiality" which at first seems contradictory, but are in fact connected and juxtaposed in order to attribute a concrete and physical significance (see Figure 6 and Figure 7) to the digitalisation of architecture (Gramazio, et al., 2014). Thus, the constructive logic of programming and the material realization are linked to each other (Willmann, et al., 2013). These significant results from the synthesis of data and material made possible by the robotic fabrication processes. In all, the material's properties are digitalized and are used as constraints and input for the digital fabrication process, which can be seen in the images below and notably how sand and the sculpting thereof helps fabricating intricate patterns for concrete wall elements. These robotic fabrication (see Figure 8) processes are a part of the term, digital fabrication. It relates to how digital tools are no longer limited to design; it becomes operative for construction processes. By making a direct connection between design data with physical construction procedures, novel design processes based on strategies of fabrication emerges (Gramazio, et al., 2014). An example of the use of digital fabrication are the Gantenbein Vineyard Facade (see Figure 9). Here the robotic technology was used to individually position and align each brick. This intricate and delicate arrangement, in regards to angles and offset, is too complex for a human to build by hand.

Current Practice

For our preliminary empirical study, where our goal was to observe and acquire an understanding of the use of robotics for form exploration, we got in contact with two students from Aarhus School of Architecture. One studying on Master level and the other

Figure 10: *The ABB IRB120 (Left) and the ABB IRB6620 (Right) Articulated arms. The ABB IRB120 has a reach of 58 centimeters, compared to the ABB IRB6620's reach of 2,2 meters (ABB,2015 - IRB120 and IRB6620). The picture does not depict relative sizes.*

a Ph.D. student in the field of architecture. Recently, Aarhus School of Architecture invested in industrial robots as a reaction to them being a big part of the future of digital fabrication and materiality in architecture. Nine of these robots are of the type ABB IRB120 (see Figure 10) which are small desktop-sized robots with a payload of 4 kg. In addition, they have one ABB IRB6620 (see Figure 10) with a total payload of 150kg. As these are relatively new at the school, not many are comfortable enough to use them, or perhaps even has a hard time imagining the capabilities and possible use-cases of the robots. However, as previously mentioned, we got in contact with two students whom are using the robots for form exploration in order to discover new ways of designing and articulating architectural design and structure.

Design process and Form Exploration

Former experience in the use of robotics in the field of gastronomy and food, revealed to us how sequential and tedious the work with robotics could be. An object or curve/surface has to be generated, with which the robot is supposed to generate a toolpath – path that the outer part, i.e. the end-effector, has to follow. This given object or surface is then translated into a collection of target points forming a path and in between positions for smoothness of motion. Furthermore, this series of target points were then converted to RAPID code in ABB's RobotStudio (ABB, 2015) in accordance to the applied tool, to create a toolpath. The toolpath would be the guiding path that the robotic arm would follow and move along, manipulating material. But first, you needed to export and save the file to an USB drive and then load it on the physical pendant of the robot controller. First then, you are able to execute the given program and observe it being performed in a physical space.

In case you have made a mistake regarding the scalability of the movement or failed to account for some other unknown factors, you would have to stop the program and then fix whatever parameter you got wrong. Afterwards, re-compile and re-execute the program on the robot controller to see the new result.

This method of offline-programming can be very tedious and time consuming, especially in the context of form exploration where continuous experimentation is essential as you want to continuously see your result variations in physical form. Thus, online-programming and the concept of what-you-see-is-what-you-get(WYSIWYG) is of more interest, e.g. like Word, where you instantly see the result of your actions by typing. The current practise resembles coding of HTML, where a page is being coded, rendered to a webpage to see the page, then recoded afterwards again and re-rendered to see the changes.

The described method of offline-programming above was basically identical in the observed session with the students from Aarhus School of Architecture, however additional software tools were used compared to our past experience.

In the two architects' work, all paths were designed using Rhino and Grasshopper (an algorithmic modelling tool for Rhino) (Grasshopper, 2015) in a virtual space, which was then controlled by the HAL plugin (HAL Robotics, 2015) for Grasshopper. This basically means, that they could change parameters in HAL and watch a simulation in the 3D viewport in Rhino – and then make further adjustments in HAL after watching the simulation. Next step was then to export

Figure 11: *Students measuring the distance between the end-effector and the surface, which is used to offset the toolpath in the software.*

it as RAPID code for testing in RobotStudio, which serves to check syntax and collisions based on a virtual replica of the physical environment. This environment is solely based on raw measurements of the robotic mounting platform and work-area platform size. After the code has been tested in RobotStudio it can be loaded to the robot controller for execution. However, as we observed inaccuracies between the model of the environment and the physical work table, adjustments had to be made.

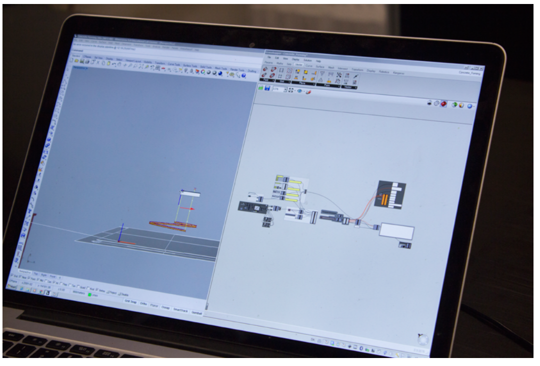


Figure 12: *The measured distance is used as a constant for the parametric design tool. After this have been changed, the code has to be re-compiled and executed again.*

The problem above, is especially exemplified by Figure 11 and Figure 12 where the two students have to continuously measure the distance in physical space between the robot's end-effector and the surface of the table. Following this measurement, they had to change some minor fault in a Grasshopper component and then re-compile and re-execute the program on the robot to see whether the change solved the problem. If not, then they would have to keep re-iterating in order to achieve the desired setup. This particular problem, leads us to the first important lesson learned, which is that a disconnect between the virtual programming of the robot and the physical space exists. Thus, the digital and physical world does not match in terms of contextual information. This kind of disconnect, could cause jumps in the design process, e.g. when the table was not aligned properly, in the environment they had to change the position of the path by adjusting a parameter in Grasshopper and then do the last sequence of steps again (see Figure 13) in order to fix it. Therefore, you could argue that these small conflicts in the use-experience interrupts and prolongs the workflow as it limits the time of discovery and restricts actual robot use.

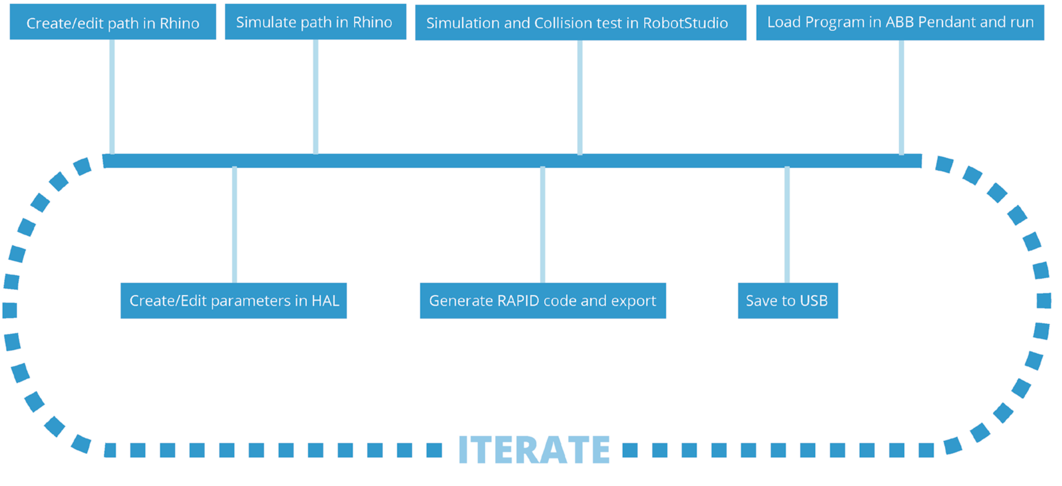


Figure 13: *A visualization of the two students' work process, as observed at Aarhus School of Architecture.*

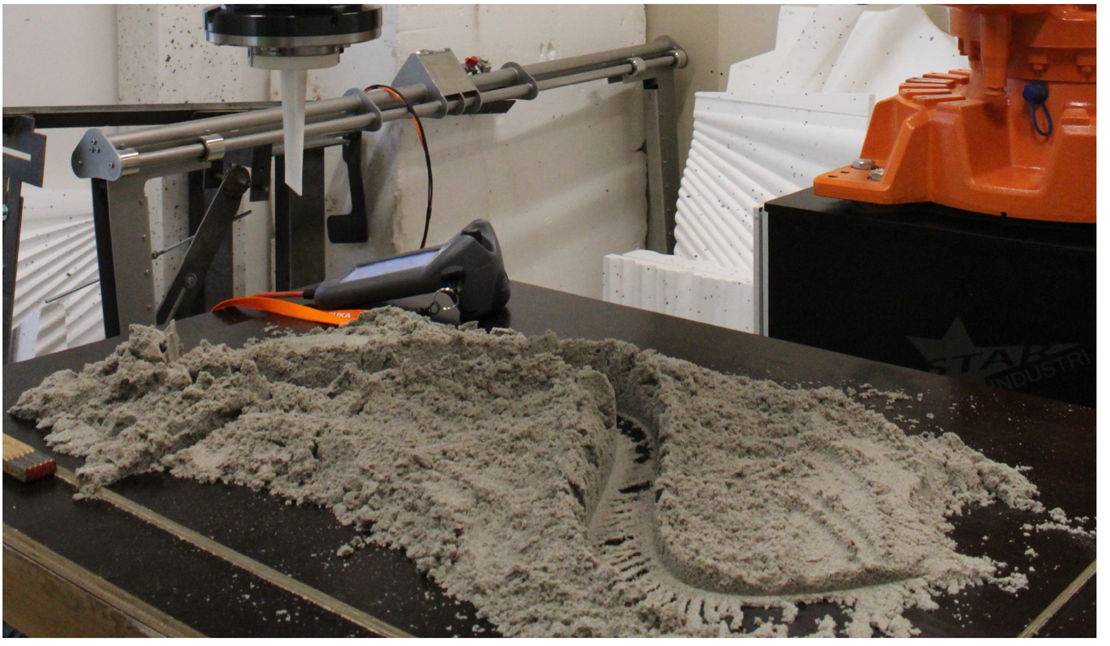


Figure 14: *An example of form exploration. In this case, the robot creates a curved "cleft" shaped as an "S".*

The two students had been working with form exploration as a method to imagine and define new shapes of structures for fabrication. In this case, various shapes and forms(see Figure 14) were tested in sand, and if the architect was satisfied with a given shape, he would then replace the sand with a fast setting concrete mixture and redo the designed shape in the given material. The process could be further extended by 3D scanning the concrete form in order to reshape and experiment with it in a CAD program, such as Rhino (Robert McNeel & Associates, 2015), to make it structurally more precise for eventually being fabricated for a construction project. Thus, the robotic form exploration is a part of a larger design process.

Interview Findings

Although we had, prior to the interviews and observation sessions with the Masters student and Ph.D. student, knowledge of some of the use-scenarios of robotics in architecture, new perspectives emerged during the observations and interviews. These perspectives and insights helped informing some of the issues and problems that average and novice users of robotics within architecture could be dealing with. These perspectives include themes such as, usability, efficiency, complexity, learnability, and iterative nature of use.

First of, there are the obvious positive subjects to touch upon, such as how an architect can integrate the fabrication constraints better in the actual design process. Much like digital fabrication is occupied with how the whole process of creating something from idea to construction ready material is embedded in the way you work as an architect with the robot. In addition, it is argued by the Masters student, that through the use of the robot you are not describing the end-result (the product), but more like describing the process of getting to an end-goal. The use of the robotic platform for form exploration is to explore form through various materials, and not to create a specific object of interest.

Additionally, both argue that the robot enables you to manage a huge field of complexities regarding the design and construction within architecture. The robot's benefits of speed, precision and repeatability are paired with the architect, makes the architect able to explore new ways of thinking architectural design. It also brings back the role of the architect to the way it was in ancient history, where the architect was the master builder that in cooperation with a stone mason, would create structures in opposition to the blueprint practise of modern times. The Masters student uses the term digital craftsmanship in this regard, to describe how the architect, through the digital tools available in the shape of a robot, are capable of constructing structures not possible before without construction specialists. This adds a whole new perspective to how architects can be more iterative in their process of creating something new and extraordinary as the robot enables the architect to imagine a more connected digital and physical world. The possibility of real world feedback informs the design space of the architect. This, however, brings us to some of the shortcomings in the current practice as presented by the interviewees along with our own observations. One of the first things that was brought up, which was despite both of them believing that the digital and physical world of architecture is more connected with the use of the robot, there is still a clear disconnect in the design process. They indicate that more feedback from the physical into the digital world would make it easier to adapt real world constraints into the digital space drawn from:

"[...] I'd love to have more feedback from the reality into the computer because you can simulate all that stuff in the computer that looks exactly like in the real world [...] so the software seems to be doing quite good, but the other way around it is more difficult." (Appendix 1)

Thus, being able to more accurately program and build the desired instructions for the robot. In conjunction with this, they say that all the focus from digital fabrication and manufacturing processes on precision etc., could be toned down to lower the entry barrier and make it easier for the average architect to approach this kind of technology regarding the way the robot is being programmed and used. This is further exemplified by the statement:

"[...] the time it takes to learn these tools isn't really available to normal architects, I would say." (Appendix 1)

Another important aspect is the sheer knowledge of the capabilities of the robots as tools. The general awareness is not present when it comes to, what the robots can be used for, as there is no clear understanding of how it is used like a laser-cutter, water-jet or CNC router, as illustrated by:

"[...] but with the robot, it comes with no tool, it comes with no software almost, so you have to look at it and think: 'why should we use it?'" (Appendix 1)

These tools are specified to some basic tasks, whereas we can see with the robot that it can be used to an infinite number of tasks. Digital fabrication tools, such as the 3D printer or lasercut involves the generation of a file that the machine can read after which it does the required work for completion. These tools also have more constraints in the form of the work-area, restrictions of movement and what materials they can manipulate, thus it is easier to think of possibilities through the confines of their limitations.

SUMMARY OF EMPIRICAL FINDINGS

Some important points have been presented and discussed throughout the chapter. From the former study, we learned how the dimension of control between human and robot can help describing the collaborative nature of tasks. Furthermore, shifting the degree of control between human and robot can provide unexpected results, thus the creative process can be affected by the robot. In addition, the robot-supported food experience study provided us with insights of how the perception of the robot can change based on common attributes such as speed, sound and acceleration.

Additionally, the observations and interviews presented some important lessons regarding the design process within the domain of robotics in architecture and form exploration. The current workflow was not optimal, as the programming process caused lots of breakdowns and was time-consuming. Consequently, these breakdowns would cause the iterative flow to slow down considerably and cause additional iterations, affecting the overall experience of the use of robotics.

The process of programming also created a relatively high entry barrier, while in addition the very nature of appearance of the robot made it hard for novices to imagine use-cases for the robot. This can be seen in contrast to other digital fabrication methods such as CNC milling, 3D printing, and laser cutter, which have different sets of constraints and specific use-cases. Lastly, there seem to be a disconnect between the digital and the physical world, which affect ones imagination and understanding of how the digital will translate into something physical.

4

Human-Robot Collaborative Creativity

In the following chapter we frame the domain by presenting an overview and definition of creativity, human-robot collaboration(HRC) and human-robot interaction(HRI). We start by defining creativity and the creative process, which creates the foundation for our adapted model of a design process. Using this iterative model allows us to discuss and investigate creativity within the field of robots in architecture. In the subsequent sections we will look at how theories of human-robot collaboration and human-robot interaction can form an understanding of the possible scenarios within the field of architecture, especially in the context of form exploration. The following theories and concepts will help guide our definition of a preliminary framework for the development of exploratory design prototypes.

.

CREATIVITY

In order for us to properly study how robots can collaborate with humans in creative processes, in the field of architecture, we start by describing creativity, both as an individual trait, but also as a sociocultural construct. Explaining and defining creativity helps us to understand how an architect can incorporate robots in creative processes and how these processes can be influenced by the actions of a robot. Current research in creativity is a multidisciplinary effort studied by e.g. sociologists, psychologists and anthropologists. We present some of the theories and definitions from this research field and relate them to the field of human-robot interaction and collaboration.

In order to define the creative individual, as a person working together with a robot in a design process, we give a historical overview of the three waves in the field of creativity research and subsequently use this to frame creativity in the context of architecture.

The first wave of creativity research began in the 1950s and 1960s, the main focus was to study the personalities of exceptionally creative people – creativity was seen as a trait of certain individuals and the goal was to determine the traits that defined the creative personality (Sawyer, 2012). In the second wave, in the 1970s and 1980s, the focus shifted to a cognitive approach, where researchers studied the mental processes that occur whilst engaging in creative behaviour. This differs from the earlier approach, as the focus is on mental processes shared by all people, thereby explaining how creativity is an ability that is embodied in all (Sawyer, 2012). A third wave built on top of the cognitive approach – the focus shifted to the sociocultural approach and cultivated a more interdisciplinary understanding of creativity (Sawyer, 2012). This greater understanding was achieved by studying the contexts surrounding the individual and how this affected and cultivated creativity.

We therefore frame creativity in architecture as being caused by the mental processes of an architect, but also the social systems, tools and knowledge that surrounds him, in turn affecting the resulting designs.

The Definition of Creativity

Based on the previous historical account, two separate approaches can be deducted as major traditions of research: the individualist and the sociocultural approach – both with their own perspective on creativity (Sawyer, 2012). We present them here in order to discuss how creativity affects the work of an architect and what can be deemed creative in the sense of created objects or design processes.

The Individualist Approach

The individualist approach focuses on studying people as singular entities isolated from the sociocultural context whilst being engaged in creative thought or behaviour, in processes such as painting, designing or writing (Sawyer, 2012). As this definition only concern single individuals, it only refers to mental processes that are associated with a single person. We adopt Sawyer's definition of individualist creativity:

"Creativity is a new mental combination that is expressed in the world" (Sawyer, 2012)

Creativity is both a mental construct, but also a physical manifestation, such as an architectural design. A creative thought has to manifest itself in the physical world in order for us to study it – we cannot study thoughts or ideas that reside within an individual. In the field of architecture, thought processes are expressed in manifestations during the design process. These manifestations include creative objects, such as prototypes, or in creative processes, such as the exploration of physical form. The most basic requirement of a creative thought or action based on this definition, is that it must be new to the creative individual – repeating an action that has already been mastered is not creative, i.e. painting the same motive repeatedly is not creative. The individualist definition is closely related to one of the oldest theories in psychology: associationism (Sawyer, 2012) - In 1855 psychologist Alexander Bain argued that "new combinations grow out of the elements already in possession of the mind" (Bain, 1855).

The Sociocultural Approach

As opposed to the former approach, the sociocultural approach studies how creative people work together in social and cultural systems – how they collectively engage in creative behaviour. Both in industry and academia, architects work together, even designs that have been made solely by one individual cannot be fully understood without looking at the broader sociocultural context surrounding him. We adopt Sawyer's definition of sociocultural creativity:

"Creativity is the generation of a product that is judged to be novel and also to be appropriate, useful, or valuable by a suitably knowledgeable social group" (Sawyer, 2012)

Novelty in this context can be judged only by a social group, who can collectively determine whether an individual creation is truly new. In the context of architecture and most creative fields, experts have internalized the criteria for judgment of their domain (MacKinnon, 1962/1978). We expand Sawyer's definition to include creative behaviour within groups of people that are interacting with a common goal, such as architects exploring form or investigating fabrication techniques. Hence, we relate the notion of group creativity to our objective of investigating creativity in human-robot collaborative teams.

The Creative Process

As presented in the earlier section, architects can engage in a creative process that can result in creative objects, such as prototypes. Sawyer have defined the creative process in an eight staged model, which is generalized to the act of creating (Sawyer, 2012). The eight stages of a creative process can be seen below:

1. Find the problem
2. Acquire the knowledge
3. Gather related information
4. Incubation
5. Generate ideas
6. Combine ideas
7. Select the best ideas
8. Externalize ideas

(Sawyer, 2012)

In relation to the field of architecture, the most interesting stages in a design process, are stages 5-8, whereas ideas can be shapes, forms or structures. Generating new shapes in form exploration, combining it and externalizing them through fabrication of full-scale models for buildings. Therefore, we limit our scope to the creative process during these stages, where a robotic collaborator can intervene and interact with a human and vice versa. This narrowing of our scope is grounded in the argument made by action theorists in the sense that creative ideas often happen while you are working with your materials, such as the moulding of clay contributes to the sculptors' creativity (Sawyer, 2012).

A more design-oriented model of the process has been made by designer and researcher, Karl Aspelund, who have conceptualized and divided the design process into seven stages (Aspelund, 2015). However, he states: *"[...] the structure should not be taken for the real world. The reality of the process is not quite so linear and clear" (Aspelund, 2015)*. The stages are as follows:

1. Inspiration
2. Identification
3. Conceptualization
4. Exploration/Refinement
5. Definition/Modelling
6. Communication
7. Production

(Aspelund, 2015)

We focus on stage 4, exploration/refinement, with emphasis on exploration. In this stage, designers explore and experiment with concepts through various visualization methods, such as sketching or modelling. We see a relation between Sawyer's creative process model and Aspelund's design process model, as Sawyer's stages 5-8 can be implemented within the Exploration/Refinement stage of Aspelund's model,

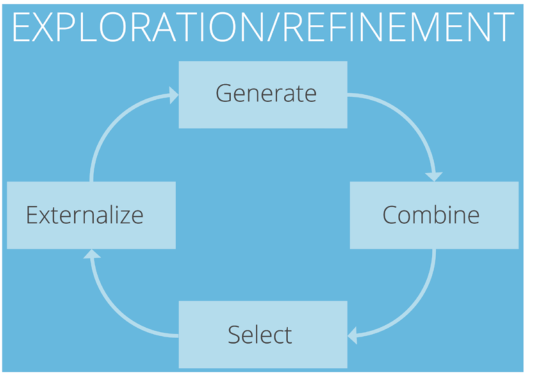


Figure 1: *Our adaptation of Sawyer's creative process model and Aspelund's design process model.*

exploring combinations of ideas through an iterative process.

In practise, this is seen as exploration through experimenting with shapes and materials, specifically form exploration, i.e. the method of exploring material characteristics and forms, as we saw during our observations with form exploration in fast-setting concrete. We adapt the relevant stages from each model and present our own model for the creative process, focusing on exploration - See Figure 1.

In summary, we narrow our investigative focus to the exploration stage of the architectural design process, specifically investigating how humans and robots can collaborate, both contributing to the creative process, hopefully resulting in a creative outcome.

HUMAN-ROBOT COLLABORATION

Human-robot collaboration (HRC) is, as opposed to human-robot interaction (HRI), more focused on the collaboration between human and robot when a common goal exists, such as in traditional human-human collaboration. HRI, which we will discuss later in this chapter, is more general and includes interaction, but limits the focus to actions that involves another human or robot. The difference between HRC and HRI is rather blurry, as HRI is required for HRC to exist. However, HRI is more concerned with the particular actions taking place between human and robot, much like human-computer interaction is interested in the actions taking place between human and computer. HRC, however, is more concerned with how we as humans perceive the sum of these actions taking place between human and robot. Do we feel that through the interaction, the robot and us, are both working toward the same end-goal? Whereas in HRI, we are more interested in how effectively we communicate, or interact, with the robot.

Although fully autonomous collaborative robots are still relatively far from reality, we see an emerging trend of collaborative control for dynamic-autonomous robots (Bruemmer, et al., 2002) (Cote, et al., 2012). Our intention is not to develop fully autonomous robots for the context of form exploration in architecture, but to sketch out a design space and framework for creativity in architecture supported by robotic agents. But first, we need to elaborate how collaboration is defined in literature between human and robot in a shared workspace, and which mechanisms of collaboration we see fit as relevant measures in the design space for human-robot collaboration and robotics in architecture and lastly, in the design process of form exploration.

We use the term, communication as a way to illustrate that for human-robot collaboration to exist, communication has to be present between human and robot. In this regard, communication covers the various modes of interaction, or as defined in collaborative systems (Grosz, 1996), communication channels. Thus, a subsection will describe how communication can take place between human and robot in a perspective of collaboration in form exploration. A final note to the structure of this section, is that even though communication is somewhat embedded and required for collaboration to occur, we have chosen to split them up in order to focus on one aspect at a time with all its' underlying themes.

Collaboration and Robotics

As we see robotic technology leave the factory floor and move into the more complex and diverse human environments, we have to consider the human-robot team, in which human and robotic agents collaborate in a shared context on shared tasks with a common goal. This perspective in HRC has been adapted from research from human-human collaboration, where conversing participants attempts to reach a shared understanding or a common ground (Green, et al., 2007).

In the following subsections, we will foremost briefly describe where the underlying theories and themes of HRC originate from. Secondly, we will elaborate on the importance of what the role of the robot takes in the aspect of collaboration. Thirdly, we will look at what common ground and joint intention means and what they mean for human-robot collaboration in general. Lastly, but most importantly, we will see how these aspects are tied together by communication. This will be seen in a perspective of industrial robotic arms in architecture with the goal of developing exploratory prototypes for envisioning the emerging ideas from these theories and concepts.

Human-human collaboration

Much research in HRC takes inspiration from the research field of human-human collaboration. Especially much work from Cynthia Breazeal (Breazeal, et al., 2005) (Hoffman & Breazeal, 2004) takes inspiration in how human teams collaborate and communicate in order to successfully complete tasks in a shared workspace setting. Whereas human-human collaboration and communication rests on multi-modality, HRC can be more limited, depending on whether the robot is a humanoid. Human-human collaboration makes use of speech, gesture, gaze and non-verbal cues. However, in many cases collaboration can also be enhanced by real objects or parts of the users' real environment (Green, et al., 2007). These attributes provide some guidelines to what a robot should have in order to effectively support human-robot collaboration.

Robotics role in collaborative tasks

Robots are finding new roles as assistants (Kwon & Suh, 2011) (Pineau, et al., 2003) and companions (Lupetti, et al., 2015) (Stiehl, et al., 2009) (Chang, et al., 2013). These roles go beyond the traditional view of robotic agents acting as simple replacements of humans in assembly lines, thus toiling away in the background. Even in industry, when relying heavily on robots, such as the automotive industry, we see an increasing shift toward a more collaborative approach with robotic agents working with humans (Shi, et al., 2012). This is also seen in the new products released by manufacturers of industrial robots, such as KUKA's LBR iiwa 7 robot (Kuka Robotics, 2014) or ABB's YuMi robot (ABB, 2015). Researchers further signal that there is an emerging and evolving culture and new attitudes toward robots in society not only as tools, but as partners in human activities and creators of culture (Sabanovic, et al., 2014) (Samani, et al., 2013) (Laursen, et al., 2015). This recent change in perception of how robotics can be used in complex human creative environments, opens up new possibilities for the use of robotic agents across various domains working jointly with humans. In particular, we can see an increasing trend within the field of architectural design where robotic agents are being considered as important entities in the design process, while they in addition can create new directions for the development of architectural design as illustrated in our preliminary empirical study.

Hoffman and Breazeal (2004) approach human-robot collaboration from the standpoint as teamwork implying a sense of partnership that occurs when agents work "jointly alongside" others instead of acting upon others (Grosz, 1996) (Hoffman & Breazeal, 2004). It is this definition of "working jointly with" that makes sense to talk about in the context of architectural design. By integrating the robot furthermore in the process of creativity and letting it contribute to the exploration of design ideas, it has the ability to become a valued partner. The robot can be seen as an active partner in the design process instead of a tool to use when a finalized design sketch is ready for production. Our hypothesis is that this will allow for a more iterative and explorative design process and create a new experience for the user, as the robot will be considered an entity reacting and acting on its own providing unexpected results for the human. In addition, the possibility of interacting with the robot during the form exploration process has the ability to strengthen the iterative nature of the design process compared to current practice.

Furthermore, Hoffman and Breazeal (2004) argues that a goal-centric view in this matter is crucial in the context of teamwork, as goals often provide common ground for interaction. Thus, while in a team, the human and robot has to obtain a common goal and a joint intention to reach that goal. Though one might argue that there is no clear and defined goal in the design process stage of form exploration other than manipulating some material; we do believe that there is an overall goal of committing to the task, in conjunction with completing it with a product in mind.

Although much of the research within HRC has been focusing on humanoids as the robotic collaborator, we believe that for collaboration to occur, the robot does not have to be designed as a humanoid in order to engage in meaningful interaction, e.g. for reaching common ground. It comes down to the use of the communication channels and how we perceive the robot in the process of creating something, which in our context of the thesis will be form exploration. These communication channels can consist of the environment, and the use and manipulation of real objects (Green, et al., 2007).

Another aspect when considering robots for collaborative tasks, is the mental models of humans in a collaborative system with robotic agents. It is necessary for the human to have an understanding of how the robot gathers data, process information and make decisions, especially in the case of fully autonomous robots (Phillips, et al., 2011). This is of particularly interest, when discussing the appearance of the robot and how people with limited prior technological knowledge approach and perceive a robotic agent. People often form inaccurate or overly presumptuous mental models about a robot's function (Phillips, et al., 2011). In addition, novice and new users of robotic agents may not know, or are able, to imagine of what use the robot really is, i.e. the non-humanoid self-expression without any clear additional hardware attachments makes it hard for a novice user to imagine what the robotic capabilities are - within an architectural setting. By adding examples of hardware attachments, such as various end-effectors, e.g. a spatula, which we will use in the form exploration examples, the user's mental model of how the robot function both mechanically and conceptually may change; thus inviting the user to imagine possible use-cases. This was especially emphasized in our interview with the two students from Aarhus School of Architecture as described in previous chapter.

Furthermore, in order for a human-robot team to effectively work together, the robot also has to reason with the humans' intentions, beliefs, desires and goals, so it can perform the right actions at the appropriate time (Hoffman & Breazeal, 2004). Consequently, the robot also has to convey information about its own set of intentions, needs and goals to establish or maintain a shared understanding and belief about the task-at-hand. In this case, ascribing mental qualities to the robot is legitimate as teamwork requires mutual understanding of the internal states of each partner (McCarthy, 1979).

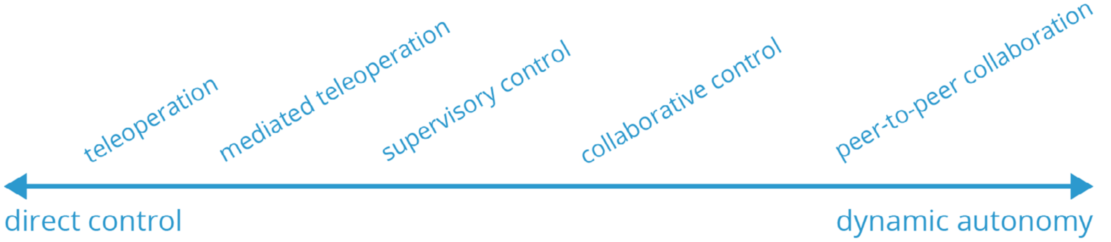


Figure 2: *Adapted model of level of automation from Goodrich and Schultz (2007).*

"To ascribe certain beliefs, knowledge, free will, intentions, consciousness, abilities or wants to a machine or computer program is legitimate when such an ascription expresses the same information about the machine that it expresses about a person." (McCarthy, 1979, p. 1)

When reaching consensus that the robot is actively providing you with feedback in a creative design process of form exploration, you can see as the quote above indicates, ascribe certain human aspects to the robot. This in turn, have the capability to make you perceive the robot as more than just a tool, thereby reaching an understanding of it being a partner working jointly with you on a task of form exploration. The nature of perceiving the robot as either a tool or a partner, can be further exemplified by the dimension of interaction between human and robot as seen in Figure 2 based on earlier work (Parasuraman, et al., 2000) (Kaber & Endsley, 2004) (Goodrich & Schultz, 2007). Figure 2 gives an overview of the spectrum of the level of automation that we envision to work within when developing the exploratory prototypes. This model depicts the relationship between the role of the robot and the interaction. We introduced in our history review, how the Spirit rover corresponds to dynamic autonomy and the Robonaut (Goza, et al., 2004) (Hoffman & Breazeal, 2004) corresponds to the direct control dimension. The model could be extended even further to encompass full autonomy (such as the iRoomba cleaning robot) (iRobot Roomba, 2015) as well, but since this is out of the scope of the thesis, we do not find it necessary to elaborate on this aspect.

Common Ground and Joint Intention

As indicated in the section above, there is a need for a human-robot team to acquire common ground. In any collaborative setting and interaction, a central feature is the establishment of common ground, which is defined by Clark as *"the sum of [...] mutual, common, or joint knowledge, beliefs, or suppositions." (Clark, 1996, p. 93)*.

This means that it helps collaborators to know what information their partners need, how to present this information, and whether they have interpreted the information correctly.

Although common ground theory originates in the research of understanding the conversation and collaboration between people and not between human and machine, early research has extended it to human-computer interaction (Brennan & Hulteen, 1995) (Paek & Horvitz, 1999). This research suggest that we can improve interfaces by thinking about the user's experience as a conversation, in which to develop a shared meaning between the user and the machine interface (Stubbs, et al., 2007) – we extend this to cover the context of the human and robot as well.

More recent research suggest that you achieve common ground in a team by communication devices (or channels). These include gestural indications, obvious activities, or salient perceptual event (such as alarms, visibly flashing lights etc.). This means that for a team to reach common ground and consensus of what each team member is going to do, information needs to be communicated in order to successfully engage in joint activity. This can be achieved through various communication mediums/channels (see Figure 3).

Joint Intention Theory (Cohen & Levesque, 1991) predicts that *"an efficient and robust collaboration scheme in a changing environment requires an open channel of communication. Sharing information*

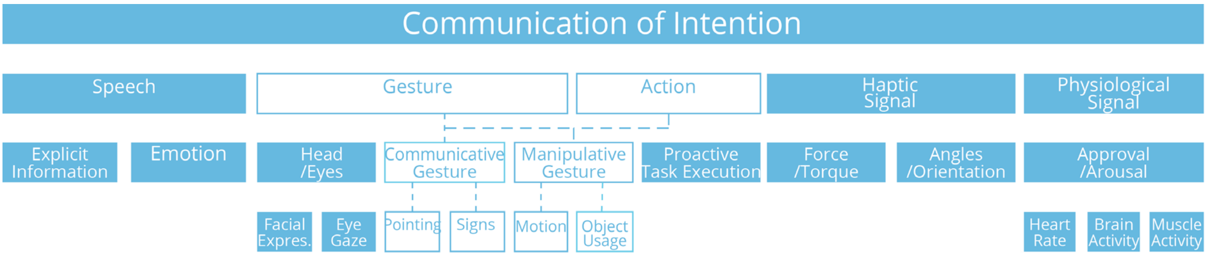


Figure 3: *Adapted figure of communication of intention showing main ways of communicating intention in HRC. Only the layout has been changed from the original. The areas of interest are highlighted as white boxes. (Goodrich and Schultz, 2007)*

through communication acts is critical given that each teammate often has only partial knowledge relevant to solving the problem and different capabilities, and possibly diverging beliefs about the state of the task" (Hoffman & Breazeal, 2004, p. 3).

The statement above indicates the importance of communication between human and robot, at all times, in order to support efficient teamwork.

This leads us to the next section, in which we will present some key theoretical aspects of current practice in communication within human-robot collaboration and what we can apply to our future exploratory prototypes for establishing common ground and reach a sense of collaboration in an architectural research setting of form exploration.

Communication

As introduced briefly earlier on an important criterion for collaboration, is the ability of the participating entities, in this context a human and a robot, to be able to communicate. This communication, or interaction, happens through communication channels which is needed to reach an understanding of common ground as indicated earlier. Our focus on communication will be in the context of a non-humanoid robot and what we imagine is useful in order for the human to perceive collaboration between human and robot in the domain of architecture.

Communication of Intention

hus, to reach joint intention, the collaborators have to effectively share their own intentions and beliefs. It is the human, in most cases, who sets a goal and therefore has an intention to reach that goal. This is where the robot needs to effectively estimate and recognize the human's intention, which can be shown by either deliberately by explicitly or implicitly communication or actions. Beneath, we have an adapted figure of communication of intention from Bauer et al (2007).

The interesting in the classification of communication of intention in Figure 3 for our initial work, is the two categories gesture and action. We find them particularly interesting as they can help us toward a solution for the disconnect between the physical and digital world in terms of programming and controlling the robotic agent. In addition, these categories can be applied to a non-humanoid robot, as they can function as an environmental communication channel, where the human and robot communicate via the environmental setting, i.e. through the direct manipulation of objects.

Returning to the two mentioned categories of communication of intent, we find communicative and manipulative gestures, which is an emerging approach in human-robot interaction and collaboration (Bauer, et al., 2007). Additionally, it is a focus toward non-verbal communication, which in recent studies play an important role in the coordination of joint activity in teams of human and robot (Breazeal, et al., 2005). It builds on already existing knowledge about human-human teamwork where verbal and non-verbal communication plays a significant role when coordinating joint activity.

As our work is done with a non-humanoid, we focus on the non-verbal nature of communication. Breazeal et al. (2005) argues that the role of non-verbal behaviour in coordinating collaborative behaviour for physical tasks in a shared workspace, do have the capability to be more effective, in addition to being a more efficient way of interaction in HRC. While much of this can be tied to humanoids, since they have a larger variety of means of communicating non-verbally by gaze direction, nods etc., we believe that non-verbal communication with a focus on communicative and manipulative gestures has a role for robotics in architecture. This is because most work within architectural design is being conducted in a shared workspace between human and robot.

Hence, it makes it interesting to further explore and investigate the possibilities of non-verbal communication between human and robot in the process of designing and programming the robot. This could be manifested in more direct manipulative gestural communication between human and robot. This could, in turn, serve to connect the physical and digital world in a more natural way for the architect compared to current practice of programming and execution. In conjunction with above, it seeks to lower the entry barrier for architects without prior programming experience, which we learned from our interview with the Masters student and the Ph.D. student, could be a problem for new users. In addition, a common language has not been developed as of yet between human and robot in architecture, which further reduces the possibility of negotiation. This forces the architect to rely on the robot as a trustworthy and active partner in a collaborative setting. Thus, as indicated earlier, the architect has to function as a resource that serves the robot, providing information and processing.

Strengths of Non-Verbal Communication

When describing non-verbal communication, it is important to distinguish between implicit and explicit communication. The two does not necessarily rule out each other, but it is important to acknowledge their individual strengths and weaknesses. Earlier work (Breazeal, et al., 2005) has been looking into how non-verbal communication can affect the human-robot teamwork regarding efficiency and robustness. Where explicit non-verbal communication is the most studied and is defined by the authors as deliberate communication, where the sender has a goal of sharing specific information with the partner, implicit non-verbal communication is at its infancy. An example of explicit non-verbal communication can be that the robotic agent could choose to point at a specific object of interest, or, if a humanoid, the robot could nod as a response to a human's query. The implicit non-verbal communication is more restricted to the form of information that are not deliberately communicated. That is, the observable behaviour of the robots' internal state. This could be the human who reads the gaze direction to infer what and where the current attention and interest of the robot is or the robot registers what object the human is currently working on, thus reacting in an appropriate way.

Furthermore, studies of body expressions in non-humanoid robotic agents have also shown the strengths of how much information, and in some cases emotions, can be communicated in a human-robot collaborative environment (Novikova & Watts, 2014). This often applies to the speed and acceleration of the robotic movements, where the perceived nature of the robotic movements, such as aggressive behaviour, can be derived. This indicates that the motion of the robot itself can function as a strong communication channel providing informative cues for the human observer.

Communicative and manipulative gestures are a form of non-verbal communication. Communicative gestures are gestures with an explicit symbolic or a semantic meaning. These can, as illustrated by Figure 3, be pointing gestures or primitive signs holding complex information. This can be derived in a sense of "showing is telling" metaphor. Manipulative gestures are body and hand gestures related to actions that a person does in the environment, e.g. manipulating objects or simple motions. Furthermore, they can be of more implicit nature, although they can still be interpreted as explicit in various occasions. However, even though the person does not necessarily mean to convey information through manipulations and motions, intention can still be derived by the robotic agent. This feature of deriving unconscious intentions from actions and motions is in most cases limited to more complex AI systems, we, however, still intend to touch upon this in conceptually during the thesis.

A thorough elaboration of the more specific interaction aspect of communicative and manipulative gestures will be presented in the coming section, as part of the more descriptive nature of interaction methods in human-robot collaboration.

HUMAN-ROBOT INTERACTION

As introduced in the previous section of HRC, human-robot interaction is concerned with the actions that takes place between human and robot, with a focus on understanding, designing, and evaluating robotic systems. In regard to this, interaction requires a communication channel between the human and robotic agent just like for interaction to happen between human and a computer requires a communication channel, or even human and human. In the previous section we discussed some of the modalities and aspects of communication, with a focus on how collaboration can be supported. Additionally, in the HRC section we introduced the concept of level of autonomy, which we described via Figure 2, while referring to our earlier work presented in Chapter 2. In Goodrich and Schultz (2007), they present the HRI problem and what attributes affect the interactions between human and robot. These attributes are as follows:

- ***Level and behaviour of autonomy***
- ***Nature of information exchange***
- ***Structure of the team***
- ***Adaptation, learning, and training of people and the robot***
- ***Shape of the task***

(Goodrich & Schultz, 2007)

These five attributes, which include the aforementioned levels of autonomy, have a significant role in defining and designing for human-robot interaction. We believe that the nature of these attributes form and shape the current design space of human-robot interaction. We will refer back to these attributes later in the thesis, as the foundation for the framework of human-robot interaction and collaboration in architecture.

In addition to the section in HRC about the robotics role in collaborative tasks, Goodrich and Schultz adapts and extends Scholtz's taxonomy of roles that robots can assume in HRI. These roles are as follows:

- ***Supervisor: A supervisor role can be characterized as monitoring and controlling the overall situation.***
- ***Operator: The operator is situated to modify internal software or models when the robot behaviour is not acceptable.***
- ***Mechanic: A mechanic must be co-located as the interactions regarding this role will be focused on the physical nature of the robot platform and work area.***
- ***Peer: This role assumes "face-to-face" interactions between human and robot, where each contribute skills according to their ability.***
- ***Bystander: This role is the most limited regarding interaction, as is mostly concerned with co-existing in the same environment.***
- ***Mentor: The robot takes on a teaching or leadership role for the human.***
- ***Information Consumer: The human does not control the robot but retrieves information from the robot.***

(Scholtz, 2003) (Goodrich & Schultz, 2007)

Direct manipulation will be a returning term, which we will use to describe an overall intend in the form of interaction. The term was originally presented by Shneiderman (Shneiderman, 1983) and covers the concept of how physical objects, in our case, can be used to translate digital commands to actions, instead of relying on a syntax and semantic matter. As an example, the use of a keyboard, mouse, or joystick to move a visual cursor. This is in contrast to having to convert physical commands into the correct syntactic form to be compiled (Shneiderman, 1983). This particular aspect correlates with lessons learned in our initial empirical work, in which we identified breakdowns in the process caused by the lack of immediate respond to your actions (in this case the programming of the robot). Furthermore, the use of the concept, direct manipulation can help achieve the system in question, becomes enjoyable to use, easy to learn, and users can gain confidence and mastery, because they are directly initiators of action (Shneiderman, 1997).

In the following section we define features of communication in a HRI perspective. Additionally, it is relevant to talk about human-robot interfaces as well in the sense of interaction, as interfaces sets the boundaries for the interaction. They also provide the necessary means for the design of the interaction strategies for the particular system in mind.

Interfaces

Natural User Interfaces (or NUIs) have had an emerging interest in the recent decade, e.g. commercial products like the Wii Remote (Nintendo Co., 2015), Microsoft Kinect for XBOX (Microsoft, 2010), and Leap Motion (Inc., 2012) - where the interaction modes of in-air gestural and body-tracking are prominent. As NUIs aim to provide a seamless user experience where technology, or the perception thereof, is invisible (Jain, et al., 2011), it has the capabilities of providing an interface, and thereby interaction modes that makes the user's experience of using robotic technologies feel more natural – i.e. the interaction with the robotic agent does not correlate poorly with how one would do the task-at-hand otherwise. In this case, the architect should be provided with modes of interaction with the robot that supports his or her normal type of work within the given context of architectural design.

In order for the nature of the interaction to be more relatable for the novice user of robotic technologies in architecture, tangible user interfaces (TUI) suit this perspective well. Ishii and Ullmer (1997) defined TUIs as interfaces that *"[...] augment the real physical world by coupling digital information to everyday physical objects and environments." (Ishii & Ullmer, 1997, p. 2)*. It is our hypothesis that by making the programming process more tangible for the user, the interaction could become more meaningful to the user. By more tangible, we mean connecting the digital properties of programming (e.g. creating a curve or changing parameters) to either the materials or physical objects the user is able to use manipulative gestures on.

Natural User Interface

Even though industrial robots have been available since the '60s, they are an emerging and constant evolving technology in various domains outside the industry, such as architectural design. Which in turn makes them more accessible to novice users. As such, the natural user interface seeks to take advantage of modern input technology in order to explore the experience of using emerging technology in a familiar and comfortable way (Araullo & Potter, 2014). NUIs in general, are those that enable users to interact with computers, in the case of the thesis it is the robotic agent, in the way that we interact with the world (Jain, et al., 2011). An important thing to note here, is that the term "natural" does not apply to the interface itself. The natural element is referred to in the way the users interact and feel about the given product. Hence, what they do and how they feel while they are using it (Wigdor & Wixon, 2011).

The interaction modes of NUIs are many and encompass, among others, those methods of communication of intention as seen in Figure 3. In many cases, it can be the combination of input and output that are experienced as ***natural***. These combinations are referred to as multi-modal experiences. This stems from how human's interaction with the world is multi-modal, as multiple senses are engaged, thus is part of what we can define as a natural experience. However, we have to be careful of what we call ***natural gestures***, as many gestures rely on the cultural setting and context. In addition, the mapping of a gesture and the emphasis on being natural, can cause ambiguities of how the interaction should be carried out correctly. Donald Norman (2010) argues in the wake of an example with the Wii remote and a bowling game for the Nintendo Wii, that the gestural convention was too natural as it led to unexpected, unfortunate consequences. These were where the users ended up throwing the remote because the analogy of releasing a button matched with that of releasing a bowling ball. Hence, users were throwing the remote, and in some cases, breaking the television screen. When working with a robot, aspects of the interaction with the robot should be carefully considered regarding safety as the robot is a powerful machine capable of damaging near surroundings and people.

Tangible User Interface

The field of architecture is originally based on craftsmanship, by connecting the digital to the physical world when using tangible blocks for programming, we believe it is possible to create an easy and accessible work environment for novice users of robotic technologies in architecture. This serves to make the interaction richer in a sense that it provides the opportunity to create meaning of function in the interaction. We propose that making the interaction between human and robot, the process of programming, tangible, we will then be able to couple user action and robotic behaviour in a more familiar way to the user. It could be argued that a middle way of combining TUIs and NUIs will have some interesting and positive effects to the experience throughout the creative design process, as they make it possible for the architect to engage more directly and hands-on in the interaction and design process of digital fabrication in addition to be more immersed. This showed to be true in a design context in a study by Kim and Maher (2008). In addition, they conclude that ***"[...] physical interaction with objects in TUIs produce epistemic actions as an 'exploratory' activity to assist in designers' spatial cognition." (Kim & Maher, 2008, p. 248)***. This means, that the TUI supports the designer's spatial cognition, which in turn means that the users cognitive load is reduced.

Sharlin et al. (2004) suggest that support for epistemic actions is an important factor in the success of TUIs. The strength of epistemic actions were originally explored and highlighted in the well-known study of epistemic and pragmatic action by Kirsh and Maglio (1994). They illustrated how the players of the game Tetris (Tetris, 1984) rotated the bricks to see how they would fit in the correct position instead of mentally rotating them. The use of epistemic actions were greatly increasing the decisive performance of placing the bricks. This also suggests, that a trial-and-error approach seems highly relevant which in addition corresponds well with the traditional workflow of an architect. This in turn, theoretically support our hypothesis that interaction with physical objects and making the interaction with the material and robot more graspable and tangible will support a more connected digital and physical world, in addition to reducing the architect's cognitive load when programming the robot.

Communication

Communication (interaction) can take various forms as seen in Figure 3, these are of course largely influenced by whether the human and robotic agent is in close proximity. Thus, communication can be separated into two more general concepts and categories; ***remote interaction*** and ***proximate interaction*** as briefly presented in Chapter 3. As architects, when in the context of a creative process working with a robotic agent, are collocated, the scope of the thesis will be involved with proximate interaction. In addition, as introduced in former section, one of our focal points will be on the categories of gestures (see Figure 3). Gestures can be considered as hand or arm movements, where an action is implicit or explicitly carried out. This makes sense to describe gestures as ***communicative*** and ***manipulative***.

Pavlovic et al. (1997) introduced a refined and slightly improved taxonomy of hand gestures (see Figure 3), originally proposed by Quek (1995), for HCI. This taxonomy was later adapted by Hoven and Mazalek (2010) to involve both hand and arm movements with additional properties for each category. The scope of the thesis will build on Pavlovic et al.'s taxonomy of hand gestures extended with properties of Hoven and Mazalek's taxonomy.

As seen in Figure 4, manipulative and communicative are the key aspects of gestures. We briefly introduced the concepts of manipulative and communicative gestures in our HRC section. The communicative gestures are further divided into acts and symbols as seen in Figure 4. Symbolic communicative gestures cover referential types of gestures and modalizing gestures and are used as a reference to actions. The referential gestures are exemplified by how a user might refer to a wheel by doing a circular motion with the index finger, where modalizing gestures can accompany speech, e.g. “Look at that wing!” and the modalizing gesture specifies that the wing is vibrating (Pavlovic, et al., 1997). Acts on the other hand, covers the gestures that are related to the interpretation of the movement itself. Such movements can according to Pavlovic et al. (1997) be mimetic by their imitating actions (known movements), or deictic, so-called pointing acts, which also is illustrated in Figure 3 which depicts the ways of communicating intention in HRC.

Another way to describe communicative and manipulative gestures are to map them as semantic as the former and direct as the latter, in relation to the functional control (interaction) of the robot. This mapping will later on in the thesis be referred to as the interaction type, when discussing the development of prototypes aimed at form exploration in architectural design.

As we introduced tangible user interfaces in the previous section, it makes sense to talk briefly about gestures and physicality. While gestures in 3D space can have many advances, they often lack physical or haptic feedback. When the gestures are used for manipulative purposes, as it is often seen in architectural design, which could be by changing the layout of a moulding form in sand, it can be tricky for the user to align the gesture in virtual space to what happens in the real world. By this we mean not being able to feel the material properties, the architect cannot properly adapt and react to the changing properties of the material, since the full length of the interaction is shaped by how the material reacts to the architect’s interaction. It can also be the simple way of being able to better imagine the scale at which you are working in.

In addition to the functional benefits, physical objects and the way we interact with the physical world, can have a rich variety of expressive properties, which stems from their varying forms and materials, e.g. shape, weight, texture, elasticity (Sharlin, et al., 2004). Sharlin et al. (2004) further suggest that the combination of gesture interaction and tangible interaction can inherit benefits from both fields. Tangibles have the ability to eliminate some technical issues, while leveraging the design process and being less

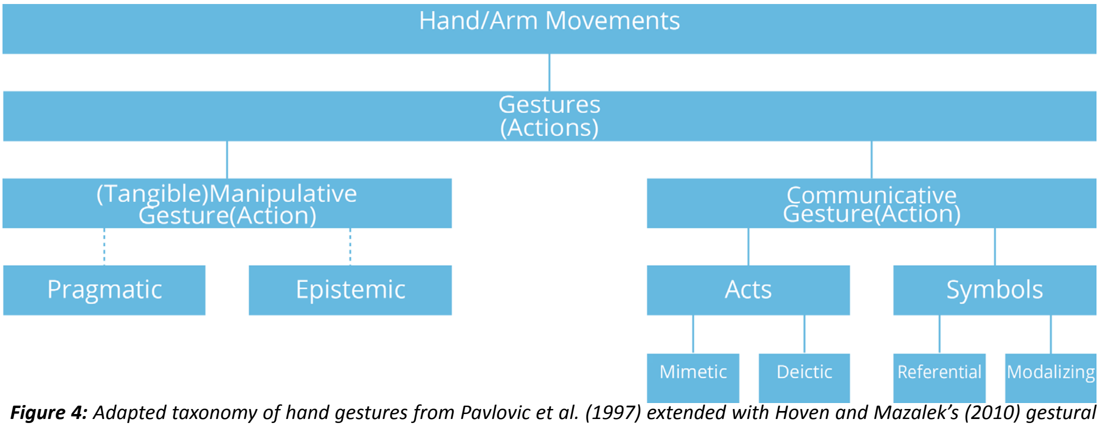


Figure 4: *Adapted taxonomy of hand gestures from Pavlovic et al. (1997) extended with Hoven and Mazalek’s (2010) gestural action categorization along with our addition of the term “tangible” to the manipulative category.*

obtrusive and more natural for the user. We will adapt these theories and concepts in order to guide the development of various exploratory prototypes in order to sketch out the design space and a framework for form exploration in architectural design in collaboration with robotic agents.

SUMMARY OF CREATIVITY, HRC AND HRI

We have introduced several essential concepts and theories that informs our understanding of the field of creativity and human-robot collaboration and interaction, and we find it interesting in relation to the domain of architecture, in particular for the method of form exploration. We identified several stages of creativity in the presented literature and theories of Aspelund (2015) and Sawyer (2012), however, our focus has been narrowed down to the exploration stage (stage 5) in the architectural design process. Within this, we seek to highlight the benefits of human-robot collaboration through the interaction methods of communicative and manipulative gestures, along with features and benefits of tangible interfaces. This is in order to connect the digital and physical world more closely for the architect and the task-at-hand; compared to lessons learned from existing practice. In addition, an objective of the thesis is to explore how, if so, tangibles can help lower the entry barrier for exploring forms with robotic agents. The presented theories and concepts will be used to explore the possibilities of developing a framework as a tool for discussing the design space regarding robotics in architecture. The subsequent chapter will present current related research in the field of robotics in architecture.

5

Robots in Architecture

In the following chapter, we present related work within the field of Robotics in Architecture that serve as a foundation for our work. First, we will look at how robots have been utilized in digital fabrication, then proceed to examine the tools that enables architects to work with robots and lastly, we present findings from two conducted interviews with domain experts from the fields of digital fabrication through the use of robots and human-robot interaction within the field of architecture. We summarise the findings up until now and frame them as questions for our research into the domain.

SOFTWARE TOOLS FOR PROGRAMMING ROBOTS IN THE FIELD OF ARCHITECTURE

In the domain of architecture, two distinct purposes exist for programming robots. In industry, robots are typically programmed for the sake of automation, as described briefly in the Chapter 3 and in research, they are programmed for mass-customization, i.e. form exploration, or one-time fabrication. By use of the word, programming, the interaction between robot and human happens through some arbitrary software tool. Offline-programming is the main method of programming robots, where the code is compiled, simulated and tested for collisions and reachability, before executed on a robot. In online programming, the robot is directly being controlled, either by a real-time interface or through physical movements of the articulated arm, i.e. teaching the robot a toolpath.

In order to facilitate the exploratory use of robots in the field of architecture, plugins have been developed that connects seemingly ordinary architectural CAD/CAM tools, such as Rhino3D and its' plugin Grasshopper, with physical robots. Plugins for Grasshopper includes the HAL (HAL Robotics ltd., 2015) and KUKA|prc (Association for Robots in Architecture, 2015) plugins, whereas the latter is actively developed by researchers Sigrid Brell-Cokcan and Johannes Braumann, the founders of the Association for Robots in Architecture – the plugin is now being maintained for and by the Association for Robots in Architecture (Brell-Cokcan & Braumann, 2011).

Brell-Cokcan and Braumann proposes the use of parametric design software to create a new design tool for robot milling, by utilizing a parametric model for calculating, visualizing and simulating the robot milling toolpaths virtually, thus informing the process and helping architects evaluate design variants (Brell-Cokcan & Braumann, 2010). They argue that one of the main reasons behind the

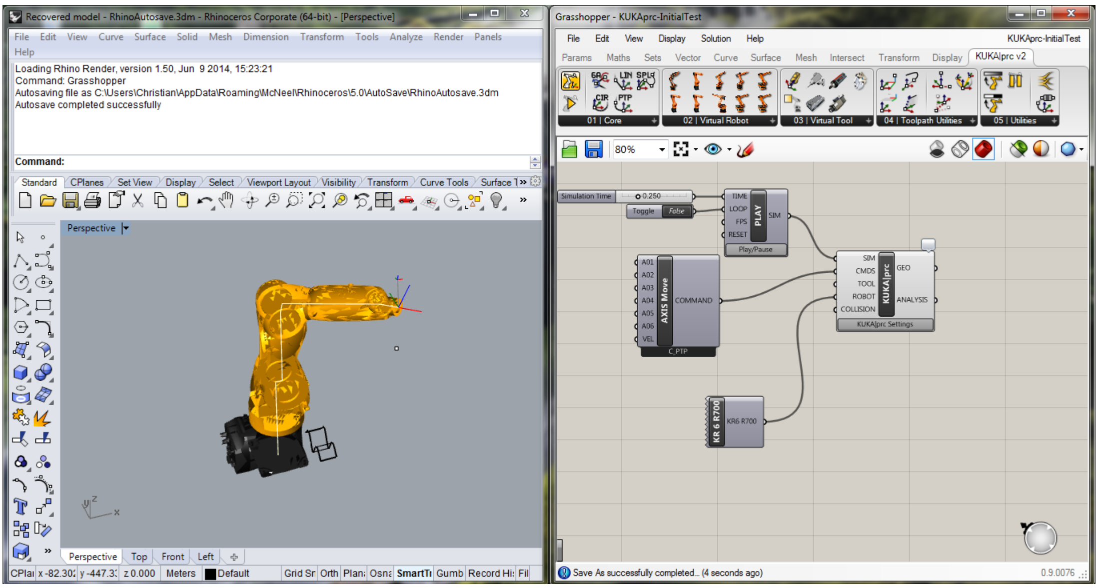


***Figure 1:** On the right side, a simple program has been made that includes the core components required for making the robot move: A simulation, Core, Command and Robot component. On the left side, the 3D viewport showcases the KR6 R700 robot in a calibrated configuration. The viewport can be used for simulations and visualising collision or reachability errors.*

development of a parametric design tool was that robots were difficult to control and had a level of geometric complexity that was hard to grasp for non-experts. They also argue that the workflow was further complicated by the necessity of controlling the robot on- or offline, creating pre-programmed toolpaths that was inappropriate for a dynamic architectural design environment. Their parametric design tool, which can be seen on Figure 1 – on the right side, allows for real-time, virtual toolpath calculations that instantly informs the architect, as he or she can see modified toolpaths digitally as they are altered.

This allows for a full simulation of the robotic movement within the CAD-software, essentially letting the architect create and evaluate desired robot movements, before realizing them. The KUKA|prc plugin is visible on the Grasshopper window, which is used in combination with the Rhino3D viewport, where a model of a specific robot model is shown on the left and the visual programming editor Grasshopper on the right, see Figure 1. Within the plugin, components are small premade code snippets, where each has a different function, e.g. one for circular motions between points, another for linear motions on a plane etc. By connecting these components, a toolpath can be parametrically created and afterwards, adjusted by simply changing numerical values like X, Y, Z or create more intricate toolpaths by creating curves surfaces within Rhino3D and make KUKA|prc translate them into toolpaths.

The KUKA|prc plugin is much similar to HAL, which is the software used by the master and Ph.D. students from our observation in Chapter 2 – one difference is that they are using a premade, virtual template that have the physical mount and work area of the robot rendered to exact measurements of the physical work area. However, lessons learned from observations and interviews in Chapter 2 can also be applied here, as the workflow is more or less the same.

RELATED RESEARCH IN ARCHITECTURE AND ROBOTICS

Robots and the use of them in the field of architecture has been an emerging area of research and design (Brell-Cokcan & Braumann, 2010) (Brell-Cokcan & Braumann, 2011) (Gramazio, et al., 2014) (Gramazio & Kohler, 2014). Current research has had a techno-positivist and narrow defined instrumentalist view. Such approaches lack, according to Mahesh Daas (2014), critical and humanistic reflection, which is necessary to properly contextualize new technologies, such as robotics, within architecture. There is a need for comprehensive taxonomy of robotics in architecture to guide the future development of robotic systems for use in an architectural setting for multi-faceted design and/or research exploration (Daas, 2014).

Towards a Taxonomy and Framework for Architectural Robotics

Although research in HRI has, for some time, delved deep into taxonomies and frameworks. These taxonomies and frameworks are to some extend transferable to the field of architecture, however, they do not fit directly into architecture as such. Recent work from Mahesh Daas (2014) has been looking into defining and proposing four frameworks, which help to classify and categorize different ways of approaching robots and robotics in the context of architecture (Daas, 2014). We will now present these frameworks, as they sketch out important aspects of robotics in architecture and the design thereof. In the upcoming chapter, we will summarize on important aspects from both the fields of HRI and HRC along with those of robotics in architecture. However, we will now turn to the proposed frameworks from Mahesh Daas (2014). Similar to Scholtz's classification of roles that the robot can assume in an interaction perspective, Mahesh Daas's first framework is about the role of the robot in an architectural context. The framework suggest that the robot plays a significant role throughout the design, building, operation and construction process. In addition, it is assumed that the robot play an instrumental role, mediating role, and a utilitarian role in the process of design or construction of buildings (Daas, 2014). The framework consists of four categories (see Figure 2); A. Robots for design, B. Robots for fabrication, C. Robots for operation, and D. Robots as buildings.

Whereas frames A, B and C focus on robotics in architecture, frame D suggest that the distinction between robots and architecture could be inverted or blurred, i.e. buildings becomes robots for living in, which at first seems as a utopian view. However, the idea that robotics can be incorporated in the actual architecture itself probes interesting new scenarios regarding the use of robotics as something that transcends the "robot as a tool" view that dominates the current understanding of robotics.

Secondly, the kind of relationships that involve robots, humans and architecture, when considering the interaction perspective, are proposed in framework 2 (see Figure 3). In this framework, we see a connection between the taxonomy of Scholtz's role of the robot and the level of autonomy as introduced in previous chapter. The point being in this framework, is that the robot do not exist in isolation; rather, it is situated in an environment in constant interaction with one or more elements, which have to be taken into account for a holistic approach for integrating robotics into architecture (Daas, 2014).

The third framework (see Figure 4) is based on Vitruvius framework of utility, firmness and aesthetics/experience. This can be related to the concept of how people perceive and understand robotic agents, from its' practical usefulness to the more intricate delicacies of the perception of behaviour.

Lastly, the fourth framework (see Figure 5) is connected to the appearance of the robot and how it is designed and constructed. Here, we have the biomorphic, mechanomorphic, polymorphic and amorphic categories. The form of the robot is argued to be a characteristic that cannot be ignored, when examining the human-robot interaction. Our work, however, is not concerned with multiple forms, and therefore, this part of the framework will have a minor influence in our work. With that in mind, however, we do discuss the appearance of the robot in the context of how movements of the robot are perceived, as it can affect the way we think of collaboration and the way we experience robotics as a co-located entity working alongside ourselves.

Approaches to the use of Robotics in Architecture

The use of industrial robots in architecture has been dominated by two distinct approaches – the first attempts to solve practical problems with use of common engineering methods without compromising the design scope (Bechthold & King, 2013). The second approach is more tied to our approach and the scope of this thesis, as it is dominated

Framework 1: Role of robotics in the architectural design process			
A. Robots for design	B. Robots for fabrication	C. Robots for operation	D. Robots *as* buildings
Desktop or industrial robots used in the design process, to inform the design process and prototyping	Industrial robots used for bespoke or mass-customized manufacturing or deployed for in-situ or off-site construction processes	Autonomous or semi-autonomous robotic assemblies such as building skins, and other components	Dwelling in a robot with extensive mobility, autonomous or semi-autonomous agencies

Figure 2:

Framework 1 from Daas (2014) showing the categories of roles of the robot in architecture.

Framework 2: Robot-human-architecture interactions			
A. Architecture	B. People	C. Robots	D. All
Robot-Architecture: Robots directly engaging and interacting with buildings or participate in the design and production processes.	Robot-Human: Robots interacting with people in architectural settings, assisting, augmenting, and facilitating usability.	Robot-Robot: Robots autonomously interacting with other robots in architectural settings. Swarms of self-assembling systems and cellular automata.	Robot-Human-Architecture: Three-way interactions involving robots, people and buildings. Essential frame to consider for robotic buildings.

Figure 3:

Framework 2 showing the dimensions of interaction between robot-human and architecture (Daas, 2014).

Framework 3: Vitruvian Triad of Robotics in Architecture		
A. *Utilitas*	B. *Firmitas*	C. *Venustas*
Efficiency, Speed, Precision, Functionality, Capacity of robots and robotic agencies	Strength, Reliability, Modularity, Adaptability, Instructability of robots and robotic agencies	Interactivity, Zoomorphism, Co-presence, Psychodynamics, Beauty of robots and robotic agencies

Figure 4:

Framework 3 showing the Vitruvian Triad of Robotics in Architecture (Daas, 2014).

Framework 4: Robots considered by form			
A. Biomorphic	B. Mechanomorphic	C. Polymorphic	D. Amorphic
Robots that resemble animals, humans, insects and other living beings.	Robots that resemble machines or embody mechanical characteristics in their form	Robots that assume different forms	Robots with no identifiable form

Figure 5:

Framework 4 considering the robot to be able to take several different forms affecting the way we percieve them (Daas, 2014).

by creative and artistic experimentation that primarily seeks to inspire and leave out practicalities and constraints of construction in the investigative design process. Bechthold and King (2013), however, proposes a third and new approach; Design Robotics, which is a strategic research method. In this proposed method, they are combining aspects of the two former worlds. The method is a type of hybrid research method that seeks to combine the bottom-up, technology driven design inquiry with the more traditional, problem-centred approaches (Bechthold & King, 2013). Whereas this does not necessarily correspond well with the very open-ended design and creative process of form exploration, the strategic mind applied suits us well with how one has to consider technological and fabrication constraints in the explorative process. This does not necessarily suggest that the research through design process is compromised by a more linear model of investigative development.

Robots as a Tool for Digital Fabrication

Current research within the field focuses mostly on the possibilities that robots offers in the creation of architectural design. Much work revolves around new ways that robots can streamline fabrication processes and how the robot becomes a connection between the digital CAD/CAM world and physical world. The robot's main advantages, such as speed, precision, repeatability and overall multi-functionality has been valued highest, as they help create intricate designs and complex patterns for digital fabrication. In order to achieve this, the field has looked at how different end-effectors, such as a hot-wire foam cutter translates virtual, 3D models into precise, physical representations and how robots can work together to create large constructions.

However, we aim to take the research in another direction than most of previous research. We seek to explore new ways of interacting with the robot along with which methods of these that support collaboration in the creative design process. The focus will be on imagining new ways of interacting with robots for digital fabrication in order to identify new procedural forms.

In the following subsections, we will present more recent research into how interaction between human and robot can be designed and constructed for digital fabrication purposes with a focus on a more investigative nature in the design process. Each subsection is an independent project that we discuss in relation to our domain and ideas as they emerge.

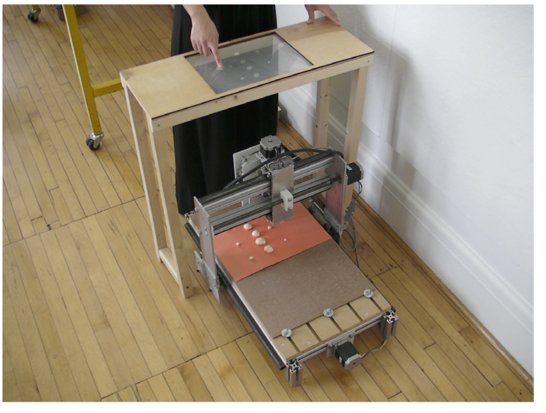


Figure 6: *Prototype of 3D printer interface. The user can do a real-time manipulation of the printed object (Willis, et al. , 2011).*

Interactive Fabrication: New Interfaces for Digital Fabrication

The project presents a series of prototypes that uses real-time input to fabricate physical form (see Figure 6 for an example). Whereas the domain of digital fabrication earlier has been costly and exclusively for the manufacturing industry, it is reaching a wider audience, as we have seen it with the students from Aarhus School of Architecture. In addition, we see that most of the current interfaces for digital fabrication are focused on the graphical user interface (GUI) paradigm (Willis, et al., 2011).

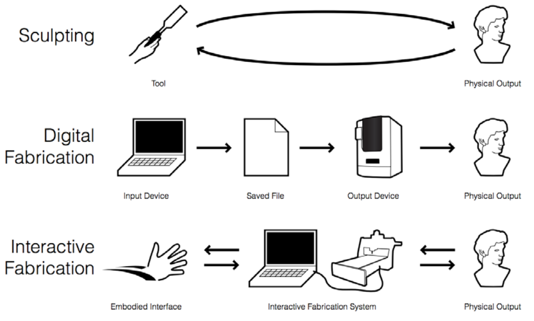


Figure 7: *Interactive Fabrication in constrast with the two existing approaches of culpting. The traditional by hand (Top), Digital Fabrication (Middle) and the proposed Interactive Fabrication (Bottom) (Willis, et al., 2011)*

As we saw it in our preliminary study, the current digital fabrication process in most cases follows a desktop publishing method: the design is created using a GUI interface, it is saved to a file, fed to an output device (e.g. USB device), and finally executed on the desired platform (e.g. articulated arm), where it is manifested in physical form. This process is very tedious and overly complex as it takes numerous disparate steps of going from a design idea to a physical form or prototype (Willis, et al., 2011). This kind of generative code-driven and algorithmic approach to fabrication are, of course, an important area for digital fabrication. However, for general design exploration scenarios it falls short, because it lacks a closer connection between interaction and physical output. A simple comparison of traditional sculpturing, normal digital fabrication processes, and the suggested new manipulative process are shown by Figure 7.

By providing users with the possibility of real-time input for fabrication processes, we believe it can open up a whole range of new creative possibilities for form exploration and experimentation. Similar to the project of Willis et al., the bridging of input and output in physicality can aid in a recapturing of the creative process. It also seeks to establish a closer relationship with the materials which allows the designer to better understand the impact of the design in the materials, thus informing the possibilities of the design space.

One of the prototypes presented by Willis et al. (2011), challenges this conventional process of digital fabrication, as introduced above, by allowing the designer direct interactive control. By using touch gestures on a translucent interface screen(see Figure 6), the user can see the physical output directly. The software detects the sketch gestures on the surface and then prints it by the 3D printing method. Authors hypothesize that by observing and interacting with the material; "[…] designers can better understand how the material behaves and 'talks back' during the creation process." (Willis, et al., 2011, p. 72). We share this hypothesis, and are guided by this for the coming development of exploratory prototypes. This view corresponds to Schneiderman's concept of direct manipulation, which we introduced in the previous chapter, where the user benefits from a "what-you-see-is-what-you-get" approach.

Another interesting aspect that the user gets from this direct manipulation approach, is that you will without a doubt get unexpected results due to, sometimes, unplanned or unconscious user-actions. This can be a negative in subtractive and additive approaches, as it cannot be reversed; however, it can in the context of form exploration in sand, provide and guide new

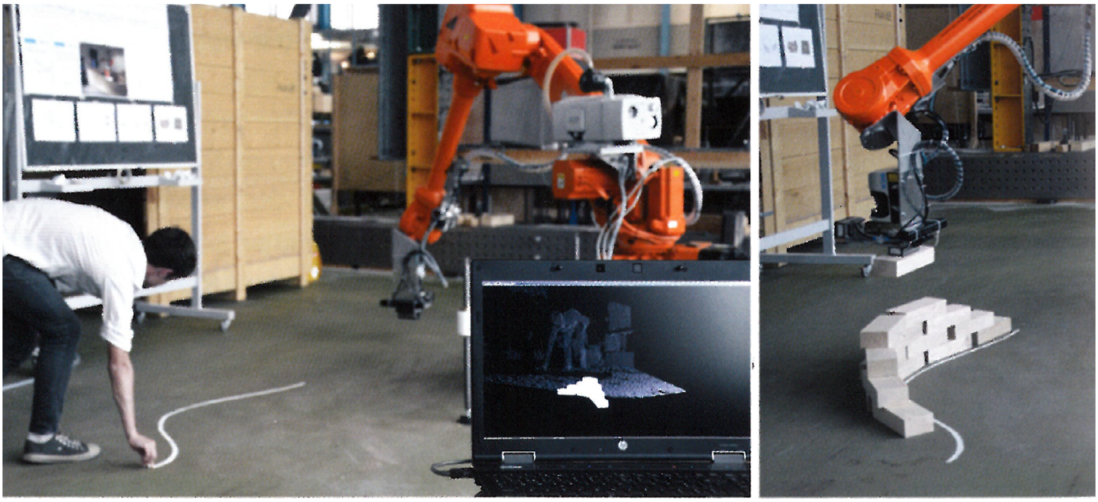


Figure 8: *The robot fabricates a brickwall along a path made by human gestures. The final development, not shown here, canbe done through a headmounted augmented reality display, which makes it possible for the architect to adjust various parameters (Johns, 2013).*

designs for the architect in question.

Augmented Reality and the Fabrication of Gestural Form

The following project revolved around the development of an augmented reality supported process of digital fabrication guided by the use of human gestures for informing the construction boundaries. Architectural design has been contributing, in conjunction with this, to technological innovation. As we have seen it with the development of the KUKA|prc visual programming tool and the Interactive Fabrication project, approaches to digital fabrication are constantly changing and improving. In this project the authors proposed a workflow, where forms (architectural design forms) are; 1) generated using skeleton-tracking and human gesture (see Figure 8) , 2) visualized, explored and modified in 3D first-person-view in situ with a see-through augmented reality headset, 3) fabricated on-site with a robotic arm (Johns, 2013).

The goal of the project was not to develop or dwell upon technology in gesture tracking or augmented reality, but to implement them as cheaply and effectively as possible, in order to explore their ability to inform the architectural design space of form generation, digital fabrication and mass customization. This way of attempting to inform the design space experimentally with e.g. gestures, correlate with the vision of this thesis. Hence, it has the ability to inform the design space by utilizing bodily movement, which in turn is translated into physical form by the robotic manipulator. Where the representation of the sketched gestures is based on augmented reality techniques, our work envisions the representational perspective existing only in the physical dimension. However, being able to preview your design sketches on a see-through display positioned in perspective to the physical world would be an interesting step further, but will at this time be out of the scope.

Interlacing

As indicated earlier, current practice of research and development in digital fabrication has been concerned with very linear production processes. However, this does not supplement the normal design process of an

Figure 9: *The human provides the robotic agent with wooden sticks and determines how it should be placed by orienting it. The robotic agent then decides where to put it on the existing structure, based on the architect's chosen orientation and where it maintain most stability.*

architect as it is non-linear. Digital fabrication and the use of advanced rule-based digital information rely on quantifiable design problems in order to be operationalized and processable by the computer (Dörfler, et al., 2013). Whereas physical tools allow for a direct sensual dialogue and connection to the material's properties. Thus, intuitive and spontaneous actions are possible. This physicality and connecting it to the input type in the interaction with the robot has, as we have seen it in above projects, been an emerging trend. The same goes for this project, Interlacing, where the overall goal is to bridge the gap between the digital and physical design tools, trying to interlace them by establishing interfaces between the physical and digital realities (Dörfler, et al., 2013). A good phrase is used to describe the necessity of intermediaries between the digital and physical, which also helps guide our research as well:

"In an environment where digital and physical realities are interwoven, physical objects become representatives of immaterial information […]" (Dörfler, et al., 2013, p. 84)

The study works with a simple case of piling up wooden sticks (see Figure 9) of varying measurements. The robotic agent, as we will refer it to in this case, stacks the wooden sticks with the goal in mind of the pile being stable. The human contributes to this process by laying out the foundation of the pile in order to form a desired shape. The robotic agent will, based on this, stack the wooden sticks following this shape. However, the robotic agent will not intervene in how the shape has been formed. The human can spontaneously control the overall shape in the way the wooden sticks are delivered to the robotic agent. Thus, how the orientation of the stick is when handled to the robotic agent.

The fact that the overall system is keeping track of the objects of interest and that the interaction is done, through physical objects, relates to the concept of direct manipulation as no external input devices are used. This corresponds well with our initial findings of how more natural user interaction could be sketched out for an architect that intends to use robotics for form exploration, as it strengthens the connection between the digital and physical world. In addition, the interplay of the human and robot in this case implies a sense of perceived collaboration as each of them is contributing to an overall goal, with individual competences.

By letting the robotic agent take some control of the design process (dynamic control) you open up the possibility of unexpected results, which an architect probably could not have accounted for in advance. Thus, new paths of the design space emerge in the spontaneous and interactive creative design process.

INTERVIEWS WITH DOMAIN EXPERTS

During our investigation of the field of robotics in architecture, we set out to gain as much insight as possible into how what current research are focusing on along with what are within the scope of future research. We arranged meetings with two of the upmost experts in the domain, Michael Knauss from ETH Zurich and Johannes Braumann founder of the Association for Robots in Architecture. We will in the following sum up relevant insights gained from the semi-structured interviews and discussions with each of them and relate it to uncovered subjects so far. The interviews were done separately and on different occasions. We will, however, present it as coupled section in order evaluate their statements in a whole.

Interviews with Johannes Braumann and Michael Knauss

Before the interview with Johannes Braumann, we attended a two-hour lecture on robotics in architecture with him and Sigrid Brell-Cokcan, whom are the authors and editors of the Springer published book: Robotic Fabrication in Architecture, Art and Design (Association for Robots in Architecture, 2013). Although much of the lecture was an introduction to how robots have been used so far for robotic fabrication, which in most cases did not bring new knowledge, some interesting statements came up, which in turn correspond with our initial findings and thoughts.

Especially the recent switch in focus on linking the digital and physical world more closely to reach the build environment. In addition to this, a strong desire, to link the actual physical workspace to the design process, exists. This disconnect between the digital and physical world, which we had identified from our initial study, was further elaborated on during our interview with Johannes Braumann:

"[...] theoretical you know exactly when is that going to be build and when is THAT[makes illustration with gestures] going to be build, and there is still this disconnect between your 3D or 4D model basically and then what's actually happens [...]" (Appendix 2)

This was in response to a question of how he envisioned the future interaction between robots and architects. This again supports the grounding for the claim or hypothesis that moving aspects of the programming out in the actual workspace or build environment, the understanding and connection between the digital and physical world becomes more clear.

Furthermore, Johannes Braumann asked a rhetoric question during the lecture of why robots were not used by everyone. The answer to this builds on the findings of how the current process is very long, tedious and disconnected from the physical world; i.e. the software is very specialized for the industry with a heavy focus on mass customization and requires a lot of programming. This, in turn, lines up with an answer during the interview with Johannes Braumann:

"[...] you only design some you know is easy to do [...] you don't know how to cope with the complexity so you then dumb down it[the design] to a level that conforms with your Grasshopper level or knowledge or whatever, which may not be the best idea." (Appendix 2)

The above sums up one of the overall problems with the current process and use of robotics in architecture. The average architects, or even expert architects, are limited in their creative process to their knowledge of programming, which is not a integral part of their skillset. This is also partly supported by Michael Knauss that states:

"[...] most of the assistants and PhD students are able to do robot programming, but if things get more complicated, it might be useful to also integrate a programmer." (Appendix 3)

However, as Michael Knauss is speaking about projects done at ETH Zurich, which has a dedicated research facility for robotics in architecture, you cannot assume that the average architect student knows about robot programming. Thus, the part of the statement about integrating a programmer supports how the complexity of the use of robots can be a hurdle.

Above leads us to how robots' role is seen in architecture by the two experts. Both researchers did not see the robot as a part of the creative process the way that we imagine it. Michael Knauss asserted it as:

"[...] the creative part is more in thedevelopment of the process and the development of the design basically [...] the moment the robot is started there's not much... not much creativity left to be explored basically [...]" (Appendix 3)

This again highlights the problem that the design has to be very detailed and planned, before using the robot, which leaves little room for exploration through actual use of the robot. The robot, however, is used in various steps of the process. Michael Knauss still says that the robot takes three roles basically; used for experimentation (which is a bit contradictory to above quote), used for one-to-one scale in-situ building/construction, and for prototype processes that is then translated into another similar technology for the final production.

However, Michael Knauss does state something interesting regarding the interplay between the robot and the human when speaking of the role of the robot:

"[...] I don't think this, the goal of these robotic fabrication processes in architecture should be full automation of building processes but more like an intelligent combination of human labour and robotic assistants [...]" (Appendix 3)

Even though they do not see or envision robots as something that is part of the creative process as such, they do however envision them as assistants rather than tools for automation of building and construction processes. Which in turn indicates that there are room for the robot taking part in more than just the fabrication processes.

In addition, Johannes Braumann opened up to how the future inclusion of a more sensor based environment so that robots also could, e.g. recognize objects that the robot could then chose to pick up or manipulate given the available sensor data.

RESEARCH PROBLEM AND SUMMARY

As we have identified throughout this chapter, there is a need to start re-imagining the use of robotic technologies, such as the industrial robotic arm. The capabilities are many, but the use of the robots are limited to the users' knowledge, in some way at least, of the available software for programming them. We believe, grounded in HRC and HRI theories, that the way we approach and interact with the robots can be imagined in new ways other than what is currently being done. Thus, the apparent disconnect between the digital world of 3D modelling and programming and the physical world, is something that can be tied more closely together. It is our hypothesis that through gestural and tangible technologies that seeks to make the programming and experimentation more manageable, this disconnect or discrepancy between the two worlds can be made less ambiguous. In addition, we seek to investigate how robotic agents can engage as active partners in the creative design process of form exploration, alongside an architect. Therefore, by engaging and affecting the creative process, the robotic agent has the possibility to provide unexpected results, influencing the design process of the architect in new ways. The above sums up the research problem guiding the thesis. We will in the following chapter, through a research through design process, present work done to explore this nature of interaction along with what collaborative elements could be used in order to change the role of the robot to take part in the creative process as well.

More specifically, we have outlined the following three research questions that scope our focus and guide our work. These research questions are addressed in the remainder of the thesis.

Research question one

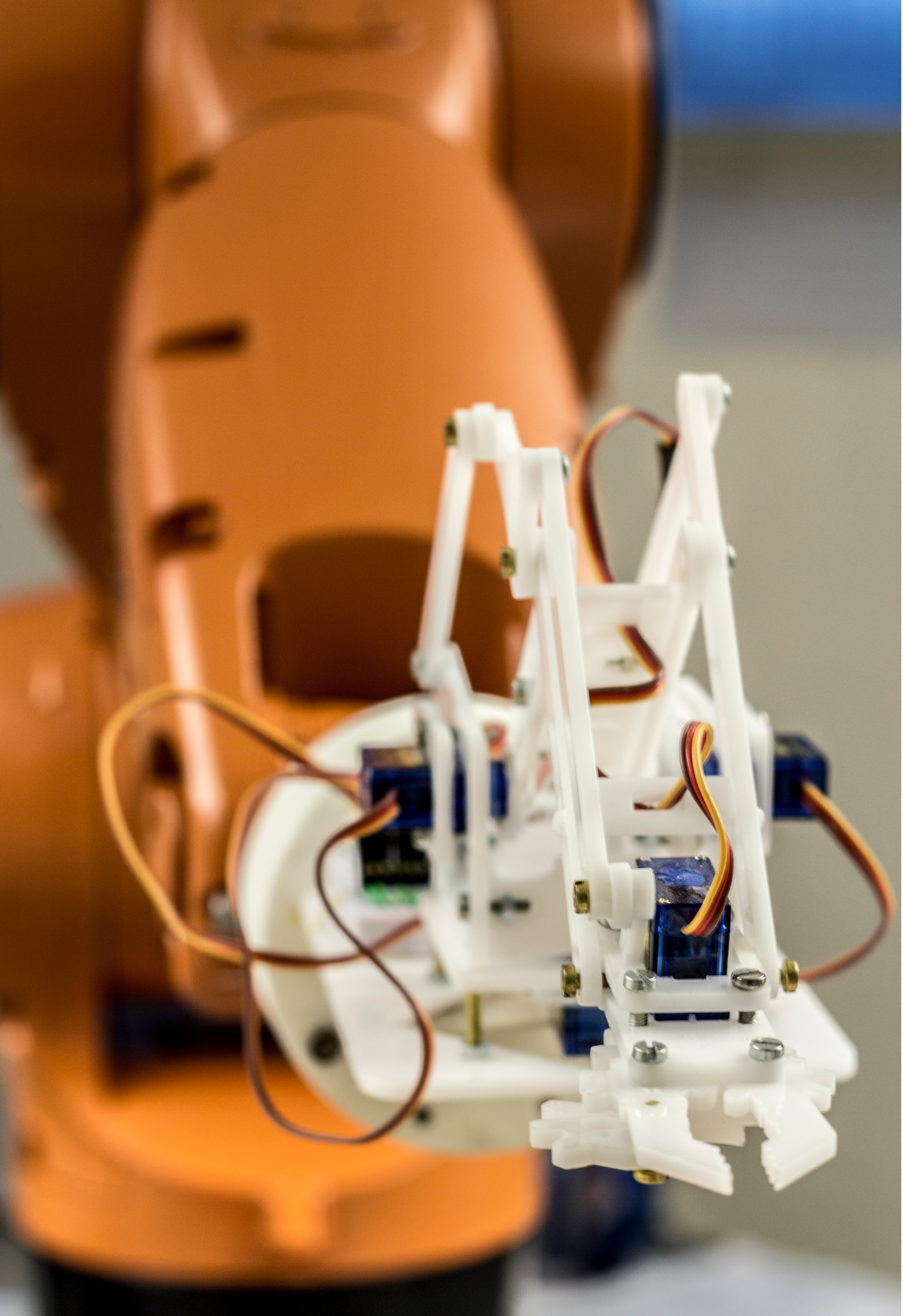
How can we design the interaction between human and robot, with the objective of improving the workflow of the architect's creative process?

Research question two

How can the disconnect between the physical and digital world be reduced, when exploring form in granular materials?

Research question three

What roles can a robot take in the activity of form exploration and how do these behaviours affect the architect?



6

Exploring the Design Space of Human-Robot Collaboration in Architecture

In the following chapter, we present our preliminary prototypes for investigating human-robot interaction, afterwards we describe our current platform for investigating human-robot interaction and collaboration in the field of architecture. We then present our initial framework for prototype exploration based on the identified key aspects of HRI, HRC and of robotics in architecture from the two previous chapter. This framework is then used to inspire and guide different interaction technologies that we have implemented on our testing platform and afterwards, evaluated through use. In the end, we conclude the chapter by presenting an experiment that works toward a comparison of the technologies used, existing practice, and how these can be further developed.

PRELIMINARY ROBOT PROTOTYPE

In the following section, we describe the technical implementation of our first prototype and present our findings in regards to exemplifying online programming and off-line programming of an articulated arm. Thereafter, we reflect upon our preliminary understanding on how novel human-robot interaction interfaces serve as an inspiration for our next prototype.

Technical Implementation

In order to quickly investigate how direct control and offline programming of the robot correlate to the dimension of control and interaction type, we built a small robot using the open-source Me Arm Robot design blueprint (MeArm Robotics, 2015). The design is similar to that of the full-scale IRB 460 industrial robot – the fastest palletizing robot in the world (ABB Robotics, 2015). The MeArm robot is built out of laser cut components made from 3mm thick white acrylic and is hold together by interlocking components and machine screws. The fully constructed robot can be seen on Figure 1.

The robot uses four TowerPro SG90 micro servos (MICROPIK, 2015) for actuation: one servo for rotation of the articulated arm, two for positioning of the arm and one for the claw end-effector. On the physical level, the servos are controllable by an Arduino Uno R3 (Arduino, 2015) and uses an external DC power supply. The Arduino Uno is running StandardFirmata firmware (Firmata, 2015) enabling software running on a PC to communicate via USB with the Arduino Uno microcontroller via the Firmata protocol. On the PC, our Arduino controller software is based on Decoded.com's leap-arduino project (Decoded, 2015) that utilizes the JavaScript Arduino programming framework, Johnny Five (bocoup, 2015), for communicating with the StandardFirmata firmware on the Arduino Uno - the controller software has been written in JavaScript for Nodejs (Node.js Foundation, 2015).

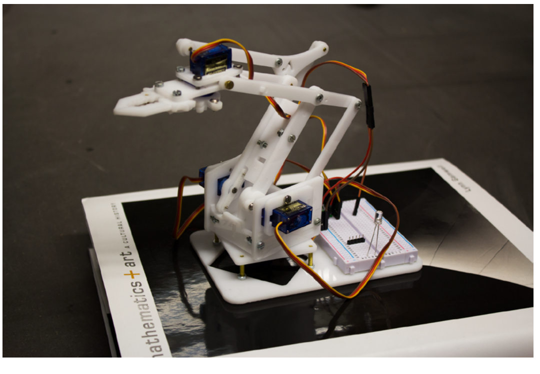


Figure 1: *The small MeArm robot including breadboard for interfacing with Arduino Uno.*

Evaluation of the robot design began after having tested the movability of the arm, with setting specific rotational values for each of the axes, thus testing different configurations of the articulated arm, the range of motion and afterwards, calibrating boundary values for e.g. the pinch state of the claw end-effector. After having noted the rotation value intervals for each of the joints, we proceeded to implement an in-air gestural interface.

We used a Leap Motion controller, which tracks the position and orientation of up to two hands using two monochromatic LEDs and three infrared LEDs (Wikipedia, 2015). The Leap Motion controller tracks the hands in 3D space within a approximately hemispherical area, as seen on Figure 2.

Our controller uses the Leap Motion JavaScript SDK directly interfacing with the Leap Motion controller, as opposed to Decoded's leap-arduino project that uses the Leap Motion web socket for receiving a continuous stream of position data. This means that our program is instead event-driven invoking the following

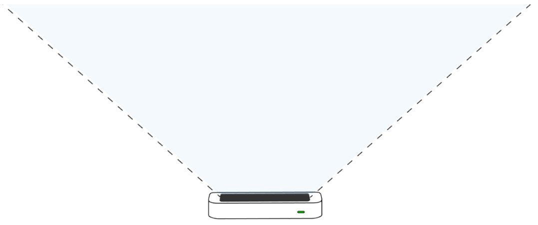


Figure 2: *The (approximately) hemispherical interaction area of the Leap Motion controller.*

function on new position data and the benefit is that the refresh-rate can reach above 200 frames a second (Leap Motion, 2015, p. Frames), which exceeds our application needs greatly.

A frame in this context, is a snapshot of position data of the users' hand interacting within the interaction boundaries of the Leap Motion. A hand object holds information about the X, Y, Z position of the centre of the palm, including information about rotation, such as the yaw, roll and pitch of the palm and lastly, the position of individual fingers. The coordinate system of the Leap Motion controller has been visualised in Figure 3.

The X, Y and Z values of the palm is mapped onto servo rotational values, as each servo can be rotated within a 0-180-degree interval and was calibrated during the construction of the robot. The mapping is first based on

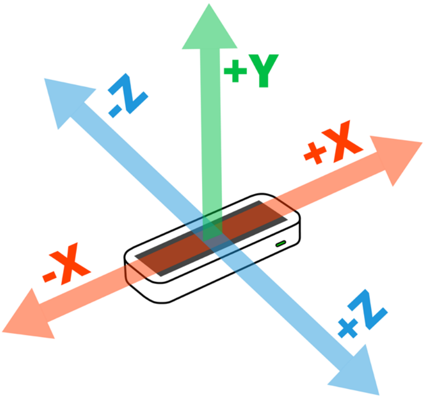


Figure 3: *The orientation of the three dimensional coordinate space of the Leap Motion.*

the human being behind the robot, facing same direction as the robot, but this proved difficult, as the end-effector could not be easily seen in some positions. Thus, we shifted to a mapping, where the human is directly in front of the robot which is then mirroring the interacting hand – the interaction is limited to one hand. The claw end-effector has a closed and open state – the state is determined by the pinchStrength value of a hand object, which is a value between 0 and 1, 1 being that a pinching finger pose, between thumb and index finger, have been recognized by the Leap Motion software (Leap Motion, 2015, p. Hand). However, due to the design of the small robot, the end-effector was always in level in relation to the surface, thus no pitch or roll data was useful for the implementation.

Evaluation of Interaction

In the following section, we highlight some of the preliminary findings in our evaluation of two types of interacting with a robot in relation to their use in the field of architecture. We began the testing of the robot using off-line programming by specifying the rotation of each servo by setting them manually:

```
a1_servo.to([0-180°])
a2_servo.to([0-180°])
a3_servo.to([0-180°])
a4_servo.to(100° (open) or 180° (closed))
```

These are executed sequentially, but the movement is performed in parallel by the Arduino Uno controller – using the above code to collectively specify the configuration of the robot arm. This enabled us to pre-program movement of the robot and state of the claw end-effector, however it also called for an iterative programming approach, as there was no virtual robot simulation, reachability testing or collision detection algorithms in place. Thus, it became necessary to test often – going from the digital environment, where we specified positions of the articulated arm, to testing it on the physical robot. As the robot moved from one configuration to another, without any intermediary points, the whole robot would wobble as it started and stopped moving. In conclusion, off-line programming is quite a time-expensive method, focusing on tweaking cycle times and planning of motion paths, which also is why it is the dominant approach for programming industrial robots through industry software such as KUKA SimPro or ABB RobotStudio.

Moving towards a more physical and connected way of interacting with the robot in online programming, where the focus is on physically teaching the robot points or simply controlling the articulated arm directly. In order to achieve a one-to-one mapping between hand and robot, we, as previously described, implemented a Leap Motion controller and utilized positional tracking of the users' hand to map directly to the position of the end-effector. Therefore, the absolute positioning of the hand is translated directly to the positioning of the end-effector, mapping the pinch gesture to the claw. The speed of movement was also sufficient relative to the mapping of the hand moving within the interacting area of the Leap Motion controller. The overall mapping was done to the rotation of each joint, thus the X, Y and Z coordinates of the hand position was translated using a mapping of the coordinates to rotational values; allowing us to evaluate quickly implement and evaluate in-air gestures. However, this could have been implemented using more elaborate approaches in regards to coordinate mapping, but this was deemed out of scope for the preliminary testing of the prototype robot.

The dimension of movement of the hand relative to the robot was also regarded to be sufficient for the first round of testing, which rendered it unnecessary to use a scale value for translating hand movement to robot movement. The delay of mapping positional data was also minimal, as the movement of the hand directly affected the rotational joints of the arm without heavy computation and done quickly without any easing, smoothing or approximation algorithms for the servos. We also note that the sound of inexpensive micro servos were relatively loud, which is highly likely to be caused by the internal plastic gears and because there was no resting/breaking mechanism for when the users' hand was stationary, i.e. letting a servo fluctuate between e.g. 159° and 160°. This lead us to wonder if hard real-time interfacing could lead to increased wear on the actuators of industrial-sized robots. In conclusion, on-line programming by in-air gestures gave us an initial experience of directly manipulating the robot, but as the robot is rather limited in its' size and strength, no real tasks could be effectively tested other than the movement of the robot itself.

The advantages and limitations of both approaches, online and offline, affects the possibilities in which human and robot interact with each other. In our preliminary testing, the only output the robot can give the human is through the movement of the articulated arm. The use of off-line programming requires the user to know exactly the steps needed for the robot to complete a task and the online programming/direct interaction allows the user to be in complete control of the robot's movement.

PLATFORM FOR EXPLORATIVE PROTOTYPING OF HUMAN-ROBOT COLLABORATION

In the following section, we present the process of building, setting up and testing our exploration platform. This platform is used as a foundation for the development of and experimenting with technical interaction prototypes – the experiments will be described later in this chapter. The experimental platform was based on a KUKA KR6 R700 sixx industrial robot which we had the fortunate pleasure of renting through KUKA Nordic AB.

The KUKA KR6 R700 sixx Robot

In order to increase our understanding through experimentation and evaluation of future prototypes, we had a need for a larger robot that truly could perform tasks that we have observed in current research within the field of architecture, including form exploration in granular materials, such as sand. We were fortunate enough to be able to rent a KUKA KR6 R700 sixx (KUKA Robotics, 2015, p. KR 6 R700 SIXX) along with KR C4 Compact Controller (KUKA Robotics, 2015, p. KR C4 compact) and SmartPad pendant (KUKA Robotics, 2015, p. smartPAD) from KUKA Nordic AB for research purposes. As described in Chapter 5, software tools, such as the KUKA|prc parametric design software are being developed specifically and in collaboration with KUKA Robotics. The KUKA KR6 R700 sixx, henceforth referred to as the robot, have six axes, a load capacity of 6 kg, a reach of 706,7 mm and is constructed as an articulated arm for general industrial production purposes. The robot can be seen on Figure 4.

Figure 4: *The KUKA KR6 R700 sixx Industrial robot shown in a surface-mounted configuration. (KUKA Robotics, 2015 - KR6 R700 sixx)*

The robot itself weighs 50 kg and can be mounted in a variety of configurations, such as on a surface, ceiling or mounted on a wall. In order to control the robot, it is connected to a KR C4 compact controller (seen on Figure 5), and a SmartPad (teach pendant) is connected to the controller for interacting with the controller and robot through a

Figure 5: The KR C4 Compact controller has an integrated powersupply for the KR6 R700 robot and handles motion and path calculations of the articulated arm (KUKA Robotic, 2015 - KR C4 Compact).

Figure 6: *The KUKA SmartPad running KUKA's own software for configuring the controller and robot. The SmartPad has a touchscreen, physical buttons and a six dimensional joystick for jogging the robot.*

touch-interface and a six dimensional joystick. All of the above was delivered disassembled on a pallet with digital manuals for setting up, calibrating and operating the robot. The SmartPad is the only method of jogging, i.e. manually moving the robot, either by using the using the six topmost buttons on the right side, see Figure 6.

Depending on the configuration, axis or Cartesian, these buttons are either used for incrementing or decrementing each specific axis by 1° or if jogging in an Cartesian system, these buttons can be used to increment or decrement on an X, Y, Z basis. The speed of which this is done can be manually changed on the lower two buttons on the right hand side, one for controlling jogging speed and one for program speed, i.e. when running an off-line programmed path of movement, referred to as a toolpath. On the right hand side, a six dimensional joystick (the black knob) can be used as a traditional joystick, but it can also be rotated clockwise or counter clockwise, and it can be pushed as a button or pulled.

Setup of Work Area

As our work revolves around human-robot collaboration, we have designed the work area as a common platform for human-robot interaction. In the following section, we describe the process of designing and building the mounting platform for the robot and how the human and robot's physical requirements informed the final construction.

At Aarhus School of Architecture, we observed that a level surface with a large enough working area to conduct form exploration was needed and the surface had to be high enough for a standing person to work with objects anywhere on the surface.

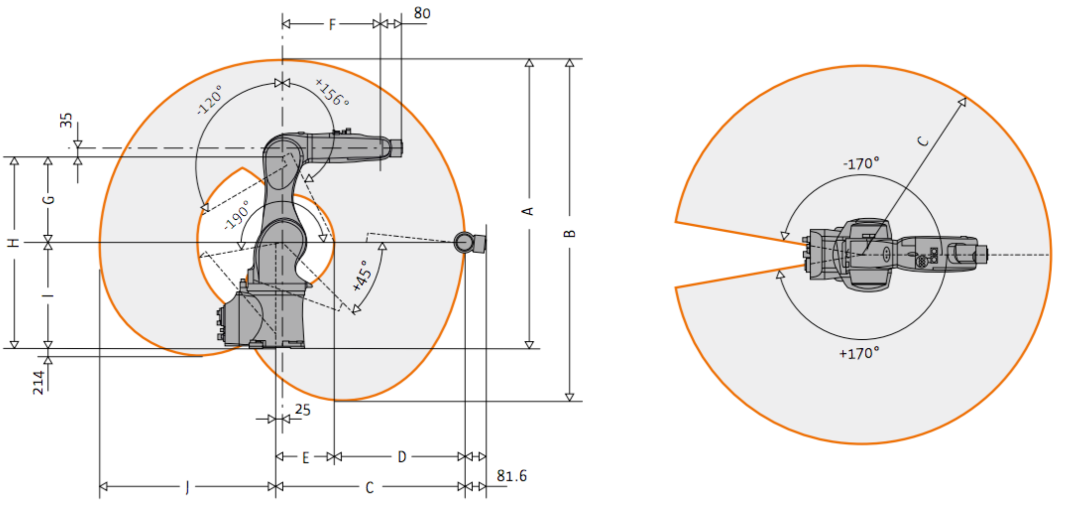


Figure 7: *On the left, the dimensions of each joint in the kinematic chain of the articulated arm. On the right, the work envelope defined as a circular sector with a radius of 706,7 mm (KUKA Roboter GmbH, 2015 - KR6 R700 sixx Data Sheet).*

The dimensions of the surface were informed by the work envelope of the robot, so that the robot would not be able to reach further than the edge of the surface. The dimensions of the work envelope and individual joints of the robot can be seen on Figure 7.

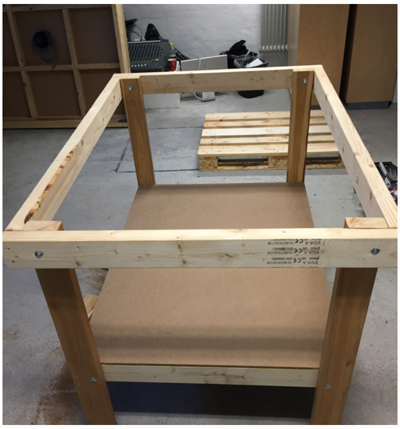


Figure 8: *The base frame for the robot table with a robust shelf underneath for holding the KR C4 controller.*

We focused on the work area in the front, half-circular work envelope, as the robot was to be mounted on the edge of the table as opposed to the centre of the surface – a lesson learned from the mapping of the 3D space of the preliminary robot prototype. This lets the surface of the table be a common ground for interacting with the object- and task-at-hand. The weight of the robot alone is 50 kg and the accompanying controller weighs 33 kg, totalling 83 kg for the basic setup without including the load of any experiments that is placed on the surface. Thus, the frame of the table needed to withstand considerable load and also required the use of carriage bolts to increase the compressive strength of the frame. As the surface had to be clear

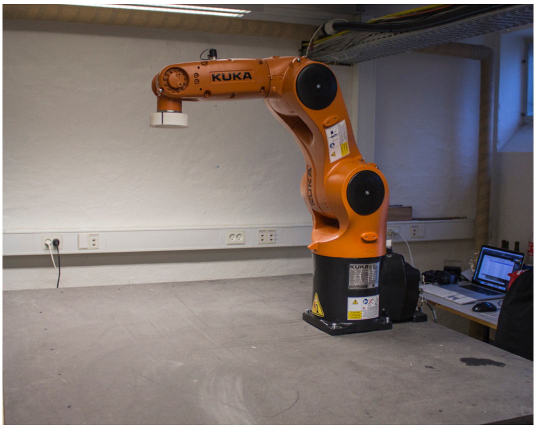


Figure 9: *The KR6 R700 sixx robot mounted on a 25 mm black MDF tabletop.*

of anything other than the articulated arm, we build a robust shelf underneath the articulated arm for the KR C4 controller, which also helped somewhat stabilize the table – another lesson we learned from our preliminary robot prototype that would begin to wobble, when moving. By extending the table top to cover more than the circular work area, the user inherently knows that the robot cannot reach any people standing beside the table.

The frame, seen on Figure 8, is made with two-by-four wooden beams, with a 9 mm thick MDF shelf underneath and a 25 mm black MDF table top on top. The robot is mounted near the edge of the surface with four M12 metric machine screws from beneath, see Figure 9. This concludes the process of building a shared working platform for robot and human. In the next section, we describe the process of connecting and configuring the robot.

Connecting, Calibrating and Configuring the Robot

In the coming section, we outline the process in which we connect the robot, and calibrate and configure it for use in the later testing stage. Connecting the robot and controller was done in accordance with the KUKA Assembly Instruction Manual (KUKA Robotics, 2014) provided with the robot, alongside the initial configuration.

Connecting the Controller

After the robot had been mounted on the surface and the KR C4 controller placed underneath, only three cables was needed to connect the controller and robot: The X30 power cable that supplies power to all motors controlling the positioning of the robot, the X31 data cable for sending positioning data to the robot and lastly, a multicore cable with a cross section of 4 mm2 for creating an equipotential bond between robot and controller. The SmartPad is plugged directly into the controller and the controller is connected to a standard 230V AC power outlet. Before turning on the controller, the battery discharge protection (X305) had to be turned off within the KR C4 cabinet. A certified electrician did a quick review of the connections and checked for grounding issues on the setup and the power outlet that was used.

Having connected all cables required allowed us to start the controller in very basic setup, however we did not have any safety circuits or PLCs controlling, e.g. a safety fence around the robot work area, which is typically seen when used in an industrial setting. Using the SmartPad, the user can shift between four different operating modes: T1 (Manual Reduced Velocity), T2 (Manual High Velocity), AUT (Automatic) or AUT-EXT (Automatic External) (KUKA Robotics, 2015, p. KR6 R700 sixx Specification). The first two modes let the user jog the robot with the accompanying SmartPad whilst holding a toggle button down as a safety measure, i.e. as a kill switch – if pressed too hard, the robot does a hard stop and if let go, the robot does a hard stop.

Calibrating the Robot

Calibration, i.e. mastering, is done on each of the six axes on the articulated arm in order to increase positioning accuracy of the articulated arm. The axes are moved into a mechanical zero position, where the actual calibration is setting and saving the value of the rotatory encoder within the specific axis on the SmartPad. Typically, an electronic measuring tool (EMT) is used to master each of the axes automatically by enabling direct feedback from each the axes' mastering ports. This lets the robot rotate each of its' axes and hereafter measures if

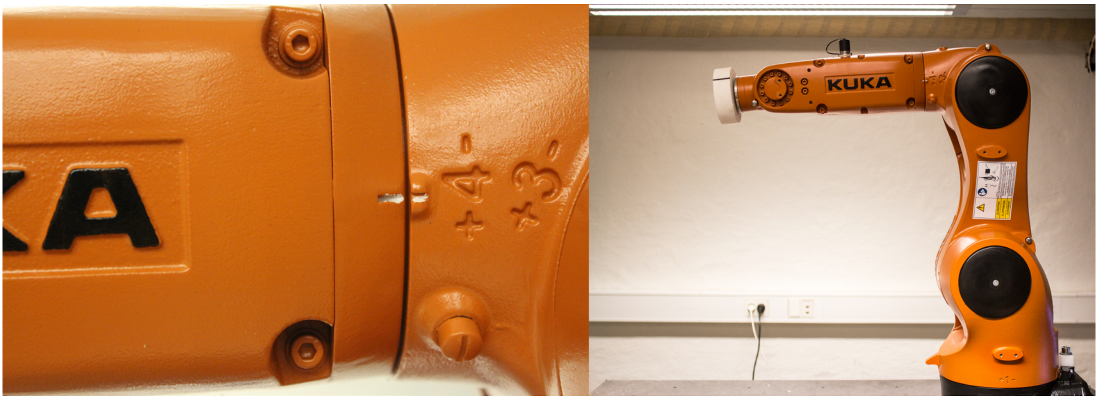


***Figure 10:** On the left, the white notches that have to be aligned (Mechanical zero position) - Notice the EMT port below the white notch on the right side. On the right, the robot configuration after all axes have been mastered.*

it has reached the mechanical zero position with the EMT. Unfortunately, we did not have access to an EMT, which limited us to doing manual mastering, which is more prone to inaccuracies. When doing mastering manually, small white notches at each axis have to be aligned, see Figure 10.

After an axis have been jogged to the mechanical zero position, the robot is calibrated through user-input on the SmartPad. Manually mastering the robot's axes introduces some level of inaccuracy in the positioning of the robot, however our use-case does not require sub-millimeter repeatable precision. After the robot has been mastered, the controller can enforce soft-limit switches, on each of the joints, that stop the robot from rotating more than physically possible, thus preventing material damage during operation.

After having calibrated six axes of the robot, the tool center point (TCP), i.e. the outer edge of a tool mounted as an end-effector on the robot's outer flange, had to be calibrated for each of the tools used in our experiments, however this was done prior to the experiments, when we attache the spatula end-effector - the calibration simply entails an X, Y, Z, A, B, C offset of the TCP relative to the mounting flange – A, B, C being the yaw, pitch and roll. As a result, the robot will move the tip of the attached tool (the TCP) to the desired position in 3D space.When using the robot, multiple three dimensional coordinate systems can be defined by the user, this could be used in cases where the robot is mounted on the floor, but the end-effector is manipulating something on a table. Therefore, a base coordinate system can be defined on the surface of the table, relative to the world coordinate system of the robot.

Configuring Safety

In the beginning, due to the controller throwing exceptions related to safety, we had to circumvent the safety features for research purposes and since a safety fence was not implemented on our platform. The security was circumvented by connecting the test outputs of the X11 connector on the controller to safety inputs, such as the Safe Operational Stop or Operator Safety Stop (KUKA Robotics, 2014, p. KR C4 Operating Instructions). The X11 connector is a port dedicated to safety circuits for implementing, e.g. safety fence with gates in assembly line productions. A picture of the X11 connector setup can be seen on Figure 11.

After having connected the test outputs of

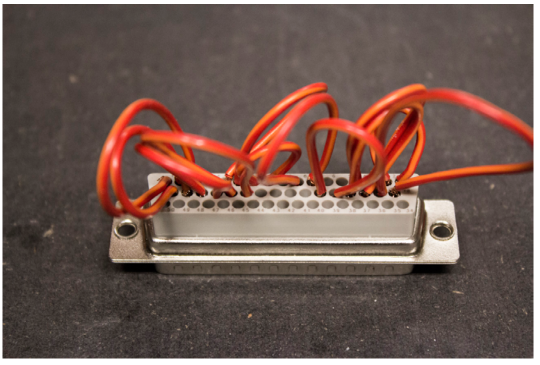


Figure 11: *A prototype plug for the X11 Safety connctor. The plug circumvents the safety circuit on the KR C4 controller.*

the X11 to the correct security channels, it became possible to jog the robot in all operating modes.

Interfacing with the Robot

After having setup, calibrated and configured the robot, we began investigating ways to interface the robot with external software tools. In its' standard configuration the robot can only be programmed by running pre-programmed paths in KUKA's own programming language, KRL (KUKA Robot Language), which will be described in segments later in the chapter. KRL programs can either be transferred via. USB-drive to the SmartPad or KR C4 Controller, or by Ethernet cable connected to the controller using one of KUKA's programming suites, such as KUKA SimPro or KUKA WorkVisual.

Third-party programs, such as the KUKA|prc plugin for Grasshopper, generate KRL program files that have to be loaded manually by USB to the robot. This is, of course, not an efficient way to obtain near real-time interaction with the robot (the authors note that it always takes three attempts to connect the USB-drive correctly), which meant we had to look at other ways to interface with the controller directly. The KR C4 controller runs two operating systems, Windows 7 Embedded (Microsoft, 2015) and VxWorx (Wind River, 2015), side-by-side and communication in between the two is over several virtual networks on different subnets. The controller has five different Ethernet ports that are used for certain tasks, such as controller bus (KCB), system bus (KSB), expansion bus (KEB), line interface (KLI) and the options network interface (KONI), where each of them have limited access to specific systems within the controller (KUKA Robotics, 2014, p. 8). The software on the SmartPad is actually a Remote Desktop Protocol (RDP) connection to the Windows 7 Embedded OS and by plugging a monitor and USB keyboard and mouse into the controller, one can logon the Windows 7 Embedded desktop.

As the robot is rather limited in interfacing capabilities, we found an open-source cross-platform communications interface for KUKA Robots, namely JOpenShowVar (Sanfilippo, et al., 2014). JOpenShowVar is based the C++ project OpenShowVar by Massimiliano Fago (Fago, 2010). JOpenShowVar works as a middleware, programmed in Java, for communicating with the KUKAVARPROXY (Fago, 2010), which is a server running on the Windows 7 Embedded environment on the KR C4 controller. Running KUKAVARPROXY (set to start on boot) on the controller also required adding firewall rules in the KR C4 software, opening port 7000 for TCP connections. The server receives data from JOpenShowVar over TCP/IP and handles reading/writing to the $CONFIG.DAT file, which holds globally accessible variables and declarations that the KR C4 controller uses for controlling and configuring the robot manipulator.

These variables can be written from client software on the computer and read in a KRL program running on the SmartPad. However, this approach does not guarantee real-time access to the underlying robot control data, but is subject to constraints – during our

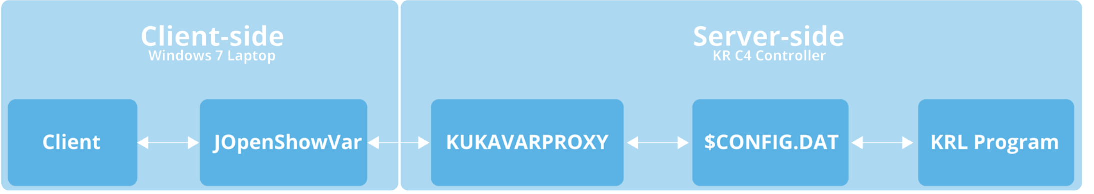


Figure 12: *An overview of the setup required for interfacing with the robot's controller. Communication between PC and controller is wireless via router through a TCP/IP connection.*

experiments, we saw a latency averaging about 5 milliseconds. In order to support the communications interface, a wireless router with DHCP enabled was setup to assign an IP address to the KLI port of the controller and to computers connecting wirelessly – the setup can be seen on Figure 12.

In the $CONFIG.DAT file on the Windows 7 Embedded environment, we have a global variable declared:

DECL POS MYPOS={X 0.0, Y 0.0, Z 0.0, A 0.0, B 0.0, C 0.0}

Which is read in our KRL Program (Snippet showcasing the most important part of the code):

ADVANCE=3

LOOP

PTP MYPOS C_PTP

ENDLOOP

In the program, we set ADVANCE=3, which is used for advancing the secondary program pointer, i.e. the block pointer that calculates and plans future motion commands (KUKA Robotics, 2010, p. 221). In our case, the pointer is advanced at least one step ahead, which enables the use of motion approximation. Within the infinite loop, we execute a point-to-point (PTP) motion command, which calculates the fastest way to move from one point to another in 3D space in the world coordinate system of the robot (as opposed to PTP_REL, which moves relative to the current robot position). The global MYPOS variable is read from the $CONFIG.DAT file continuously and C_PTP causes the end-point to be approximated using default values specified in the $CONFIG.DAT file (KUKA Robotics, 2010, p. 296). Other motion types include linear motion (LIN), where the TCP follows the a straight toolpath between start point and end point, circular motions (CIRC), where a circular movement is defined by a start point, auxiliary point and end point, and lastly, spline motions, which consists of several spline blocks that excels at following complex curved toolpaths (as opposed to approximated LIN or CIRC motions) (KUKA Robotics, 2010, pp. 239-240) – however, these motion types and underlying tool orientation can be highly customized for creating complex toolpaths. A diagram of the robot's world coordinate system can be seen on Figure 13.

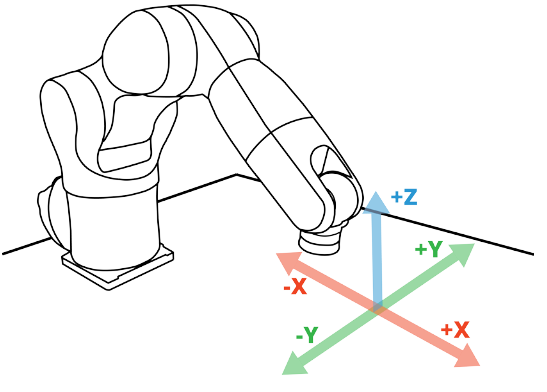


Figure 13: *The X, Y, Z orientation of the robot's world coordinate system. Yaw is rotation around the Z-axis, pitch is around the Y-axis and roll around the X-axis.*

Challenges and Reflections on the Setup Process

As a method of summarizing our experiences of the process, we discuss some of the challenges we encountered during the setup of the KR6 R700 sixx robot, in a chronological fashion.

First, it is difficult to get access to an industrial robot for use in research, unless it is in-house or enough funding is in place to buy one (including setup and training for operation of the robot) – e.g. a KUKA KR6 R700 sixx with KR C4 Compact Controller and SmartPad is around 185.000 DKK. Luckily, programs exist, where a robot can be rented for up to six months for a fee and where potential customers have an opportunity of testing it. Though, this was realised after the work on our thesis had begun. Second, the robot is made for industrial use-cases: it has a high velocity and can be extremely precise, e.g. for surface mount technology (SMT) assembly lines.

The connection of the controller, robot and SmartPad is relatively straight-forward, but requires some basic knowledge about electricity in order to ensure the equipotential bonding is done correctly and with a cable with a large enough cross-section and to ensure that the power-outlet is grounded properly.

We experienced that if the power-switch is not pressed quickly enough, letting it fluctuate between on and off, it will cause the residual-current device (RCD), i.e. ground fault circuit interrupter to switch off. This happened on two occasions and cut the power from the rest of the building using the same electrical switchboard - we want to apologize to the one person in the building, who was working on a stationary computer while it happened. We solved this partly by switching to an alternative electrical switchboard and made sure not too turn it off slowly or operate it whilst written exams were underway. We did not need high precision for our use-case, which allowed us to master the robot using a manual, jogging technique as opposed to using an EMT.

Safety is of high concern in the high-speed working environment of assembly line productions, but in our case, we needed direct access to the robot's working envelope during operation, which, of course, was a safety concern. By using the test outputs of the X11 safety connector, which is used when debugging security fences, we were able to circumvent the safety procedures of the controller. Our safety concern was partially solved by only running motions at reduced velocity in T1 mode when jogging the robot and having a hand near the SmartPad's kill switch and emergency-button (See Figure 6).

The complexity of the systems and networks running on the KR C4 controller is high, because they are hardly documented, as it is not something uneducated people should mess around with, which makes it challenging to interface with, even though open-source software exists specifically for this purpose. The network can also be hard to decipher without comprehensive documentation, as multiple virtual networks exists within the controller with different subnets and firewalls in between.

INITIAL FRAMEWORK FOR HUMAN-ROBOT COLLABORATION IN FORM EXPLORATION IN ARCHITECTURE

We have so far presented various aspects, theories and lessons learned from the domains of creativity, human-robot collaboration and interaction, and robotics in architecture. Based on our formal understanding of the rapidly developing field of robotics in architecture, we have identified some key dimensions that we wish to apply to a conceptual framework for form exploration in architectural design supported by robotic agents.

Although different attributes and taxonomies exist for both HRC and HRI, such as Scholtz's taxonomy of the roles of the robot (as presented in Chapter 4) and Goodrich and Schultz's five attributes (also presented in Chapter 4) for defining the HRC problem domain, they cannot be fully transferred to the field of architecture. In the domain of robotics in architecture, Mahesh Daas(Daas, 2014) has proposed four frameworks as seen in previous chapter which seeks to give researchers a way to design and evaluate robotic systems applied to architecture, which is a very broad field for application. However, we believe that by combining aspects and dimensions from the worlds of HRI, HRC and robotics in architectural design, one can get a better understanding of how to design for a more explorative and collaborative design process in form exploration in architectural design supported by robotic agents.

Thus, in the following section we will summarize these dimensions in relation to each other, in order to lay out the foundation for our exploratory prototypes that seek to explore and investigate the feasibility of these dimensions. Furthermore, any lessons learned during these exploratory prototypes will help us evaluate and refine the presented dimensions in order to sketch out a direction for future research into how interactivity and collaboration can support the creative design process of form exploration in architecture, whilst being supported by robotic agents.

The exploratory prototypes and the design process takes inspiration in the iterative design process model (see Figure 1 in Chapter 4) adapted and extended from Sawyer (Sawyer, 2012) and Aspelund (Aspelund, 2015). Our focus on creativity in HRC is limited to these exploratory stages defined in this model.

A Preliminary Framework

In the previous chapters, we introduced the notion that a designer can affect five attributes that affects the interactions between human and robot. These attributes, provided by Goodrich and Schultz, consisted of the following:

- ***Level and behaviour of autonomy***
- ***Nature of information exchange***
- ***Structure of the team***
- ***Adaptation, learning, and training of people and the robot***
- ***Shape of the task***

(Goodrich & Schultz, 2007)

However, we will rename some of these attributes, or dimensions which we shall call them for futures reference, to make them more understandable in addition to make them relate more to presented theories and terms presented so far. Hence, we end up with the following dimensions:

- ***Level of autonomy***
- ***Interaction type***
- ***Role of the robot***
- ***Learnability***
- ***Application domain***

The renamed dimensions above are listed in the order as the list with the original names. However, this order does not indicate to how one should approach the design of robotic systems for architectural form exploration. Furthermore, the dimension learnability will not initially be a dimension on its own. It will, however, be a sub-dimension of interaction type meant for assessing whether the system is easy to learn and comprehend interaction wise. In the following, additional aspects to these dimensions will be added and discussed in order to fully understand each of the dimensions in relation to the domain of architectural design and form exploration.

At first it makes sense to start at the application domain (see Figure 14) as this sets the scene for where the focus of the development is. In our context the application domain is within form exploration for the architectural design process. This could be other domains such as, e.g. digital fabrication. The next dimension to consider is the interaction type (see Figure 15) which contains the categories direct (manipulative) and semantic (communicative). These input types for interaction takes inspiration in the communication of intention (Bauer, et al., 2007) (Figure 3 in Chapter 4) as presented in Chapter 4, where we described and discussed the terms communicative and manipulative gestures as a way to communicate to, or interact with, a robotic agent. However, we do not limit gestures to gestures by in-air hand gestures only, but can entail gestures done through objects. Thus, we extend manipulative gestures, as presented in Chapter 4, to include the term tangible to better describe interaction with physical elements that are manipulated to communicate certain information to the robot. This also serve to cover our intentional use of TUI (tangible user interface). Furthermore, manipulative, or direct, gestural interaction also entail the use of a mouse and keyboard as objects of communicating intention. Learnability is a dimension not illustrated as a model, but will be a term that we use in the evaluation of our exploratory prototypes in order to describe the ease of use and whether the interaction is perceived as meaningful.

Furthermore, we have our proposed third dimension of the framework (see Figure 16). This dimension is adapted from authors' earlier work, the model of dimension of control and the level of autonomy presented by Goodrich and Schultz (2007). For the thesis, the layout have been adapted to suit our needs for a preliminary framework model. The original, as presented in Chapter 4, was based on a horizontal axis, where direct control was situated at the far left side of this scale and dynamic control situated at the far right. The purpose of this scale is to position it according to the dimension of interaction (interaction type). Therefore, an example could be that with a direct manipulation type of interaction in real-time, the level of autonomy would be situated on the end of the scale of direct control as the robot is controlled fully by the architect. In our interpretation of the model, we have also adapted a horizontal box model like the previous dimensions in order to get a

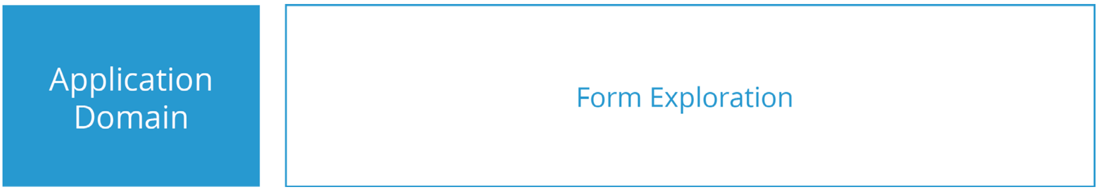


Figure 14: *The dimension specifying the application domain of our exploration. In our case, it was form exploration with robots in architectural design.*

Figure 15: *This dimension specifies the interaction type to be defined as either direct, e.g. manipulative gestures on tangible objects, or semantic, e.g. in-air gestures refering to a curve as input for the robot.*

Figure 16: *The third dimension of the framework depicts the level of autonomy. Ranging from direct control (human in full-control in real-time) to dynamic control (robot and human share control).*

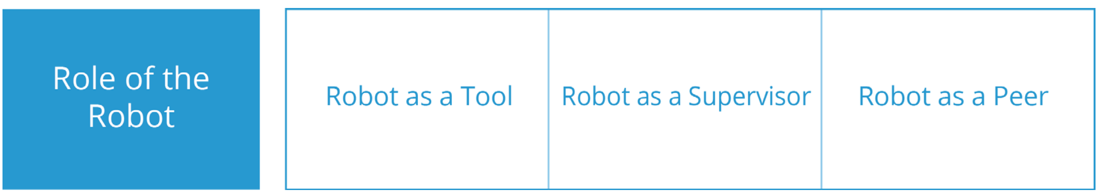


Figure 17: *Dimension of roles that a robot can assume in the context of form exploration. Ranging from the simplest, the robot as a tool for the architect. Secondly, the robot can assume a supervisor role, intervening when necessary. Finally, the robot can act as a peer, where the robot acts as an equal, sharing control in order to reach a common goal.*

consistent composition of the framework. The model is not sharply divided in direct control and dynamic control, as there are several intermediate stages, in addition to that the control can shift during the full length of the interaction stage.

The fourth, and for now, final dimension that we consider as central for the development and exploration of the domain is the the role of the robot (see Figure 17). This dimension was first introduced in Chapter 4, in order to describe how the robot could be perceived as part of the interactive environment with humans. Jean Scholtz (Scholtz, 2003) and later Goodrich and Schultz (Goodrich & Schultz, 2007) presented a taxonomy of roles of the robot which we have adapted and reduced according to the application domain of form exploration in architectural design. Thus, we have narrowed the scope of the roles to encompass the following three; robot as a tool, robot as a supervisor, or robot as a peer. Robot as a tool corresponds to the simplest way robots can be used to aid in the completion of physical tasks (Green, et al., 2007). This role is not directly inferred from a role from the adapted taxonomy. Robot as a supervisor is characterized by Scholtz as monitoring and controlling the overall situation. In the case of the domain, or context, it could mean that the robot merely observes and follows the actions of an architect and then intervenes or react, when it becomes necessary. This case could emerge if the architect tries, during a series of movements, to do something that are out of bounds of the robot's work envelope. Consequently, the robot steps in and takes control and rejects out-of-bounds movements – this is explained in detail in the next section. The last role, which we have adapted, is the peer role where the overall control shifts between the human and robot during a given task. Effectively speaking, the robot and human engage in teamwork as they are completing the task-at-hand in collaboration, by sharing the overall control, when manipulating an object or material of shared focus. This could be exemplified that the robot controls motion in the X and Y-axes, whereas the human controls the Z-axis along with pitch, yaw, roll in order to explore more complex shapes and forms.

Further inspiration for how to define the role of the robot are found in Framework 1 (see Figure 2 in Chapter 5) proposed by Mahesh Daas (Daas, 2014). However, when looking at the choice of our adapted roles, one could argue that the roles transcends this particular definition of roles. An example could be how the robot as a tool could be applied to both frame A and B (see Figure 2 in chapter 5). Furthermore, these roles are based on different application domains within architectural design, where our adaptation of Scholtz's roles are not restricted to a certain domain, but can be used across domains. However, the application domain of frame A in the framework for the role of robotics in architectural design process corresponds well with how a definition of an overall role of the robot in our context would be:

"[...] used in the design process, to inform the design process and prototyping." (Figure 2 in Chapter 5)

Thus, a preliminary framework for use to develop a number of exploratory prototypes are illustrated in Figure 18. From the top we have the application domain, interaction type, role of the robot, and level of autonomy. For a quick summary of the dimensions, we have the application domain as the first to define in what context the work is being done. In the second dimension we have the interaction type, which covers the two directions; direct and semantic, which correspond to manipulative gestures and communicative

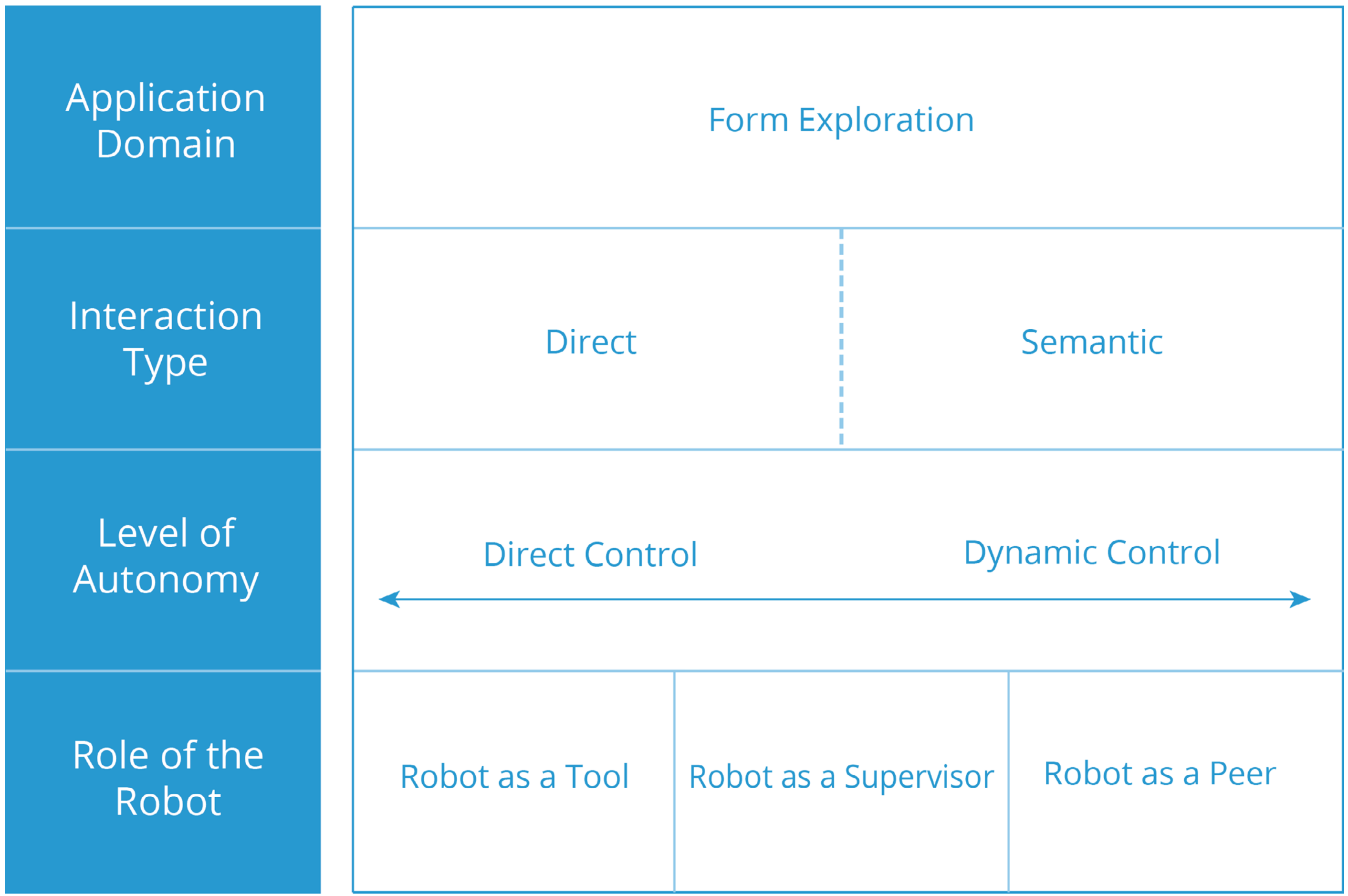


Figure 18: *The initial framework using the previously presented dimensions for describing the design and evaluation of human-robot collaboration in the context of form exploration in an architectural design process.*

gestures. The third dimension reflects the level of autonomy, which we presented in Chapter 4, where, depending on the dimension of interaction, the robot can be positioned as controlled directly by the architect or dynamically, where the control shifts between the human and robot. Lastly, the fourth dimension represents which role the robot can assume in the design process, in the context of form exploration and based on the interaction type and level of autonomy.

THE DEVELOPMENT OF THE EXPLORATORY PROTOTYPES

In the following section, we introduce our exploratory prototypes by describing their technical implementation and explaining them in accordance with the initial framework from the previous section. We evaluate how traditional HCI devices, such as the mouse and keyboard facilitates a first-hand view and a basic feeling of directly controlling the robot. We will showcase a few code snippets, then proceed to look at the interaction, through our prototype, with the robot and how this is mapped. After having accounted for the technical implementation, setup and initial use, we reflect on the level of autonomy and the role of the robot, and how the architect-robot relationship can be collaborative. In the end of the section, a comparison of the prototypes will be conducted through a small design case. Throughout the description of our prototypes, hints at future work might appear through reflections, however these are discussed in detail in the subsequent chapter.

Keyboard Interaction Prototype

Our first prototype is based on the traditional human-computer interface, the keyboard. By using the four arrow keys on the keyboard,

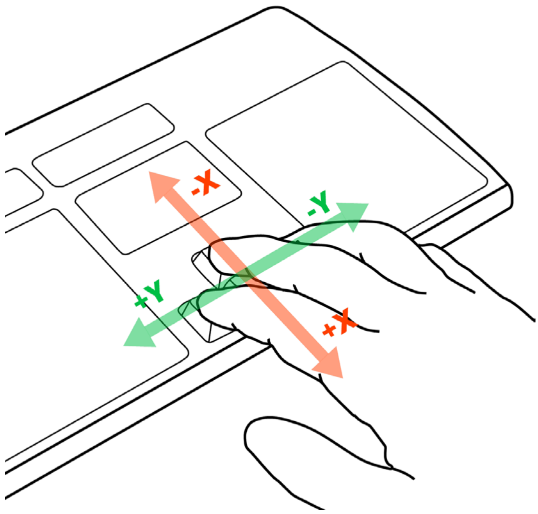


Figure 19: *The coordinate systems orientation of our KeyboardController, which is inverted relative to the robot's world coordinate system. This is based on the assumption that the architect is situated in front of the robot's work area.*

we can navigate in a 2D-space, which traditionally is the output on a screen, but in our case, we map the arrow keys to the X, Y dimension of the robot – keeping the TCP pointed towards the table surface. A short video of the keyboard interaction prototype can be found at http://cr.coel.dk (Laursen & Pedersen, 2015).

Technical Implementation

Our KeyboardController program is written in Java and uses the JOpenShowVar open-source communication interface for sending position data to the KR C4 controller, which in turn changes the position of the articulated arm.

The program implements a KeyListener (Oracle, 2015) that invokes the inherited keyPressed method, if an arbitrary key is pressed. Within this method, the program checks the KeyEvent object for details about the key pressed. In the same method, we create a KRLPos position object from the JOpenShowVar library that holds information about a point (X, Y, Z, A, B, C) in the world coordinate system of the robot.

However, by the use of arrow keys, we are limited to only populating the X and Y values for the position object. As long as a key is pressed, a new position object is generated with new coordinates and sent to the KR C4 controller. The KeyboardController maintains the knowledge about its' current position and adjusts it every time an arrow key has either been pressed once or are being continuously pressed.

Input-Output Mapping

As each arrow key either increments our decrements the X or Y value of a KRLPos object, it can only move continuously as long as a key is being pressed. Hence, the keys function as a way of controlling direction on the X, Y plane and therefore limited to movement in straight or diagonal lines – the way of controlling the robot resembles that of older 2D games that are viewed from the top. Mapping to a third dimension, Z, would have to be done with other keys, such as Page up and Page down. Control is relative to the current position of the robot, as opposed to controlling the position in an absolute coordinate system. The X and Y dimension has been visualised on the keys, see Figure 19.

One could discuss whether we should use the PTP_REL command instead of the PTP command, as the robot is always moving relative to its' current position, i.e. we cannot specify a point in 3D space with the arrow keys. However, with our current implementation, we have a boundary-check logic, which checks whether a new point is "out of bounds", i.e. unreachable or out of the work-area that we have defined. This limitation is defined and enforced programmatically on the client, subsequently, it is different from the controllers soft-limit switches on the axes. If we were to implement PTP_REL, we had to implement a POS_ACT command (not documented in the programming manual, but an internal variable that holds current robot

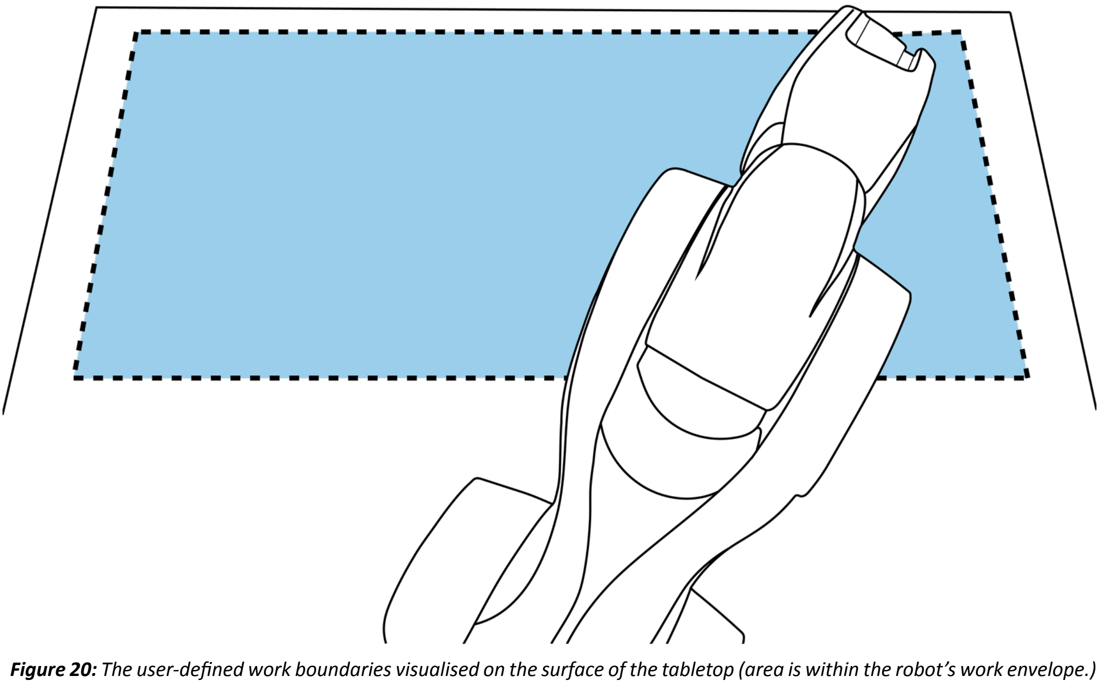


Figure 20: *The user-defined work boundaries visualised on the surface of the tabletop (area is within the robot's work envelope.)*

position and of the E6Pos Structure type (KUKA Robotics, 2010, p. 289)) for checking current position of the robot, before sending a new position command to the controller. This internal variable is only updated, when the robot is moving, therefore it becomes unwise to use this for checking whether the robot is moving out of bounds, before it starts moving.

Type of Interaction

The interaction with the keyboard is restricted to the four arrow keys and these are limited to whether it is pushed. Consequently, no velocity can be inferred by a button press, unless the key is pressed repeatedly and giving the illusion of a lower velocity due to the delays between key presses. The interaction through a keyboard with the robot is limited to real-time and pressing keys is directly determining the direction of the robot's toolpath, however limited to the X, Y plane. Pressing a single key limits movement to that direction, however pressing a button controlling movement on alternate axis, will let the robot move diagonally, i.e. pressing up and right at the same time. The interaction type for use of keyboard is direct, as it is connected to the movements of the robot in a straightforward way.

Level of Autonomy

As the robot is controlled real-time and the user only controls direction, a low level of autonomy can be programmed. The robot has no prior knowledge about the intentions of the user, if the user is planning on moving the robot in a triangle or other specific shape, thus the autonomy is limited to the current position of the robot. In this experiment, we use a programmed boundary to define a rectangular work area on the X, Y plane, in which the robot can move freely – this is done to prevent the robot going beyond maximum reach and therefore trigger a reachability exception of the KR C4 controller, causing the robot to stop moving (See Figure 20). When the robot reaches this boundary, e.g. on the x-axis, it will not move further, but instead follow the boundary, keeping e.g. X constant. This type of autonomy or robotic behaviour can be categorised as a safety precaution, both relative to the robot itself, but also the people who might be working alongside the robot at the table. The level of autonomy is mostly situated as direct control; except for cases where the user's interaction would translate into the robot hitting the boundary of the programmed work area.

Role of the Robot

Using Goodrich and Schultz's Level of Automation model, we perceive the movement of the robot as a direct translation of a button press on the keyboard, i.e. a simple method for tele operating of the robot. Thus, the robot can be perceived as a tool that an architect could use, when using a keyboard for direct interaction and in some sense, as a supervisor for the architects' manipulation of material, as it makes sure that the end-effector does not move away from the area of focus on the table.

Mouse Interaction Prototype

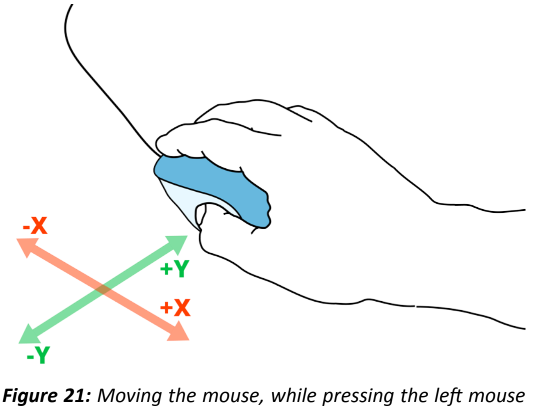


Figure 21: *Moving the mouse, while pressing the left mouse button moves the robot accordingly on the X, Y plane.*

In our next experiment, we use the mouse as a way of directing the TCP of the robot on a X, Y plane and like the former experiment, the height of the end-effector is constant. However, translating the mouse's movement on a surface and the click of a mouse button to movement of the robot showcases an alternative way of both directly moving and directing the robot to specific points. A short video of the mouse interaction prototype can be found at http://cr.coel.dk (Laursen & Pedersen, 2015).

Technical Implementation

The MouseController program is, like the KeyboardController, written in Java and uses the JOpenShowVar middleware for communicating with the controller. The program implements a MouseMotionListener ((Oracle, 2015) and inherits mouseDragged and mouseMoved methods. We implement the mouseDragged method, which fires a MouseEvent, when a mouse button is pushed and the mouse is moved. The coordinate system of the screen is offset, as the coordinate (0,0) would normally be in top left, however by adding a constant, (0,0) is in the top middle of the screen. This is done, because of the possibility of Y being negative in the robot's world coordinate system, see Figure 13. Using X and Y from the event object, we set the X and Y parameters of a new KRLPos variable that is sent to the controller, when the MouseEvent is fired, which will be done repeatedly as long as the mouse is dragged.

Input-Output Mapping

As mentioned in the technical implementation, it became necessary to offset the coordinate system of the screen, since (0,0) is in the top-left corner, but the (0,0) is in the centre of the robot's mounting base according to its' world coordinate system. As long as the mouse is being dragged, i.e. moved whilst a key is being pressed, the robot will move accordingly, thus dragging the mouse in a curve, will create a curve motion on the X, Y plane of the robot's end-effector. The motion of the input is up-scaled to the same motion executed by the robot. Scaling is a factor in this experiment, as opposed to the previous keyboard experiment, as upscaling the mouse movement can cause large toolpaths with minimal mouse movement or downscaling can increase the precision of the interaction. If the mouse is moved to an arbitrary point on the screen and the cursor is being slightly dragged, then the robot will jump directly to that point, but still keeping within a user-defined working area on the X, Y plane. In order to control the height of the robot, the Z-axis could correlate to the scroll wheel, however this would function as a different modality than simply moving the mouse.

Type of Interaction

In this experiment, the interaction is limited to moving the mouse on a surface and a left-click. As opposed to the earlier example with the keyboard, the velocity of the robot is relative to the velocity of which the user moves the mouse (within the pre-set speed-limit). As holding the left mouse button whilst moving the mouse causes the robot to move, another possibility opens up, when not holding the left mouse button down, for simply directing the robot by pointing and clicking onscreen. The factor for which the mouse motion is translated to robot motion affects the interaction as well: upscaling is useful for quickly prototyping a movement with less accuracy than downscaling, where a more precise toolpath is needed.

In the future, we could adapt a dynamic scaling factor, which perhaps could have been determined by the scroll wheel of the mouse. The point and click approach could also be further investigated as a way of creating waypoints for which the robot has to go through, consequently visualizing and creating the toolpath before executing the corresponding robot movement. The interaction type is mostly direct (manipulative gesture through an object), however, it has hints of semantic interaction (a communicative act by referring to a known shape or path, e.g. an arch/curve) as the user can gesture/simulate a curve with the mouse, which will be translated to a curved motion of the robot. A non-real-time approach could be to draw a shape; after which it would be drawn by the robot.

Level of Autonomy

At the most basic level, the robot has a safety behaviour, as the keyboard example, that keeps the robot within a specified work area. Two types of interaction can be inferred from this example, one is the real-time translation of mouse movement to robot movement; the other is a point and move type interaction, where the user left-clicks on an arbitrary point onscreen and the robot will move to the according coordinate on the X, Y plane of the defined work area.

However, the level of autonomy is mostly at the direct control end of the dimension; as it only ever takes control, when the robot needs to ensure that the architect is working within the boundaries of the work area. However, if working with a non-real time approach, the robot could choose to alter the requested shape in order to interpolate the lines and curvature etc., compared to the relatively imprecise movements directed by the mouse. This would, in turn, balance the level of autonomy as the robot takes control in some part of the interaction, therefore take responsibility of the output alongside the architect.

Role of the Robot

As the robot, in this case, does not have a high level of autonomy, which is much like the keyboard example, the role of the robot can be perceived as tool, but in some cases also a supervisor, when limiting the robot's movement to a predefined bounding box – essentially taking control away from the user on the limited axis and ensuring that the robot is kept within this boundary until it moves away from the edge again. Additionally, as we briefly exemplified in the level of autonomy section, the robot can, in some cases, be perceived as a peer, contributing actively to the design of forms and shapes by taking control and assisting in smoothing the lines and curvature etc. However, this partly depends on the point of view of the architect toward the robotic agent as an entity providing a set of skills of its' own. We use this vision for our next couple of prototypes.

Leap Motion Interaction Prototype

Moving away from the traditional human-computer interaction methods, we look at how a user can interact with the robot without an artefact, such as a mouse. We introduce the Leap Motion, which was used on the preliminary robot, for interacting with the robot and is a form of NUI based on in-air gestures. The technical details of the Leap Motion have been introduced earlier in this chapter, so in this section we focus on describing how it is implemented in our current setup and reflect on its' potential. A short video of the Leap Motion interaction prototype can be found at http://cr.coel.dk (Laursen & Pedersen, 2015).

Technical Implementation

Based on our previous experience with implementing the Leap Motion, only few code adjustments had to be made, when rewriting the JavaScript program to Java code to take advantage of the JOpenShowVar middleware. Instead of setting each axis' rotation in a sequential manner, a KRLPos object is continuously being update on each

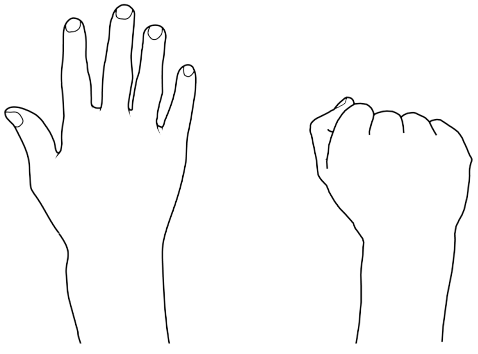


Figure 22: *On the left, the user's hand is open, facing downwards and not being tracked by the robot. On the right, the user does a grab gesture, signalling the robot to follow the position of his hand.*

frame of the Leap Motion sensor snapshot. The program checks if one hand is in the interaction area of the sensor and gets the X, Y, Z position of the palm and the pitch and roll of the hand – yaw is constant. The rotational values of the pitch and roll is mapped as well, making sure that the rotation of the hand is within user-defined limits, this is done to prevent the whole arm from reconfiguring to another position in order to reach the rotation required. The values are then checked for reachability within the user-defined boundary before being set in the KRLPos object. The KRLPos object is only sent if a closed hand has been recognized by the LeapMotion, therefore enabling the architect to position his hand in interaction space, before gesturing the robot to track him, see Figure 22. The gesture builds upon the notion of grabbing the robot and moving it directly.

Input-Output Mapping

In our preliminary robot prototype, the difference between the interaction area of the Leap Motion and the working envelope of the small robot was relatively similar in size. However, implementing the Leap Motion controller and testing it on the larger robot, we saw that upscaling the movement was required in order to have full control of the work envelope with the moderately small interaction area of the Leap Motion, see Figure 2 in the beginning of the chapter. Our experience mapping the interaction to the movement of the robot is similar to that of the mouse prototype, as a higher scaling factor, e.g. robot moves double the length as the hand, the faster a person can reach any point in the work envelope, but this is at

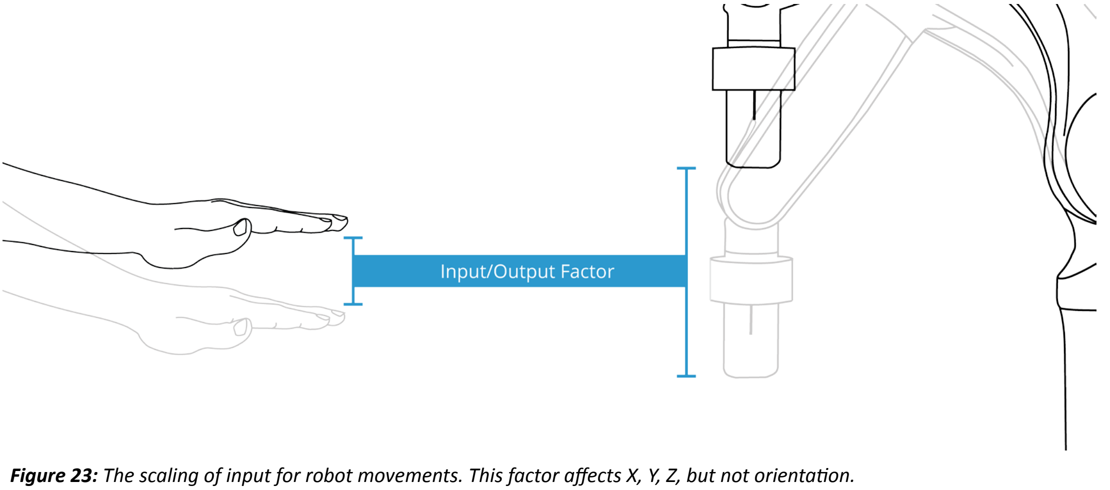


Figure 23: *The scaling of input for robot movements. This factor affects X, Y, Z, but not orientation.*

the expense of precision, which is relevant for almost any input device for computers etc. This is of course a limitation of the input device, hence the technology used for tracking the hand – see Figure 23.

When we started implementing mapping of hand rotation, we immediately saw how it became more difficult to control as the complexity of the interaction increased. This could be because of the hand to end-effector mapping, as the arm will reconfigure itself in order for the end-effector to reach a certain point translated by the position of the hand. Another way that this prototype resembles the mouse experiment, is that the hand specifies an absolute point in space that is translated to the robot's world coordinate system.

Another way that the challenge of the relatively small interaction area could be overcome is done in our RelativeLeapMotionController program, where moving the hand in a direction translates to moving the robot end-effector in that direction, i.e. specifying the direction of movement in 3D space by hand. By adjusting the resting area, i.e. area which marks "no direction", it becomes easier to stop the robot movement precisely. In our current version, we do not differentiate between left and right hand, as long as a hand is recognized, it can be used for interacting with the robot, hence by differentiating one could imagine that each hand would be mapped differently to the robot. Utilising two Leap Motion controllers could make it possible to use one hand for rotation of the end-effector and one for the positioning, consequently reducing the complexity of controlling the robot.

Type of Interaction

In the current implementation, the interaction type is primarily direct, a one-to-one positional relationship between hand and robot. In our preliminary robot prototype, we had implemented a pinch gesture for closing and opening the claw end-effector. This could be transferred to the bigger robot, however, the type of semantic interaction is dependent on the end-effector used – using a pinching gesture to power on a hotwire cutter might seem a bit far-fetched.

However, first iteration of our Leap Motion program would translate hand movement to robot movement as soon as a hand was recognized by the Leap Motion, this lead to some unforeseen cases, where the user would remove the hand and therefore move the robot in the same direction. However, after adding a grab gesture, i.e. symbolizing that the user could grab a hold of the robot, it would activate the tracking – giving the user the possibility of gesturing when control was needed and when it was not. The interaction type can be summarized as a combination of direct and semantic gestures: direct interaction for movement and semantic for controlling end-effectors or other functions.

Level of Autonomy

As with the other experiments, the safety behaviour has been implemented here as well. This behaviour is fundamental for human-robot interaction in these use-cases, as both parties know the boundaries for shared attention. This also provides the knowledge that the robot will not accidently hit a person standing beside the table top. By using the Leap Motion, the user can directly control X, Y, Z, A, B, C, but this adds to the complexity of the interaction. However, the robot and human can share control of the movement, i.e. the robot moves in a predefined path on the X and Y axes, but the user can control the rotation of e.g. the spatula end-effector, or a hotwire-cutter. The shared control is a form of collaboration, where human and robot works together in manipulating a material. This is implemented by having one of the axes, e.g. the X axis, controlled by the robot, which continuously moves in a direction, where the user can control rotation of the end-effector. The level of autonomy is dynamic, as during normal work, the robot takes control of some of the parameters of movement, whereas the user focuses on the remaining.

Role of the Robot

The role of the robot is, in the case of direct control with no robot behaviour for controlling, simply a tool for which the user benefits from the strength, speed and precision of the robot, by upscaling his hand movement to the robot. However, as seen with the earlier examples, when adding a boundary-checking behaviour, the robot takes on the role as a supervisor overseeing the work of the architect. In this example, we begin to see the third role: the robot as a peer. When sharing control of robot movement, the robot and architect is equally controlling the robot, they have responsibilities for moving the end-effector that directly affects the other partner. We see that this shared control is happening concurrently, as both parties contribute to the manipulation of the objects at the same time.

Tangible Blocks Interaction Prototype

In our tangible blocks prototype, we evaluate how a tangible user interface can create a foundation for human-robot collaboration. This marks a focus shift, from mapping input to the positioning of the robot end-effector to the toolpath mapping of the robot. We evaluate how blocks, symbolizing waypoints, can be arranged to create intricate toolpaths of which the architect has full control over. A short video of the tangible blocks interaction

Figure 24: *The tangible user interface setup on a surface for placing blocks. These blocks are tracked by the webcam mounted above*

prototype can be found at http://cr.coel.dk (Laursen & Pedersen, 2015).

Technical Implementation

Our implementation is built upon the open-source computer vision framework, reacTIVision (Kaltenbrunner & Bencina, 2007), for tracking fiducial markers that have been attached to physical objects, in our

Figure 25: *A screenshot showing the three blocks being tracked by reacTIVision.*

case, wooden blocks. For this setup, we had to build another surface for the placement and tracking of blocks, which could be setup quickly, but also hidden when testing other prototypes. We added an additional MDF plate in front of the robot; the setup can be seen on Figure 24.

A Logitech C310 720P webcam have been mounted above the new, additional table top. This webcam is used by reacTIVision. The reacTIVision framework consists of libraries for different programming languages and a standalone application that sends TUIO messages (Kaltenbrunner, 2015), which holds the state of a tangible object, i.e. our blocks, to our TUIO-enabled program. A screenshot of the viewport of the reacTIVision application can be seen on Figure 25; this is used for calibrating the camera source.

Our TangibleBlockController program implements a TuioListener (Kaltenbrunner, 2015), which inherits seven call-back methods, however we only use the following four:

```
void addTuioObject(TuioObject tobj)
void updateTuioObject(TuioObject tobj)
void removeTuioObject(TuioObject tobj)
void refresh(TuioTime btime)
```

When reacTIVision sees a new marker within camera space that have been successfully recognized, it fires the addTuioObject method. The TuioObject holds the fiducial marker id, i.e. symbol id, angle of marker in relation to camera and X, Y coordinates. We create a new KRLPos variable and set the TuioObject X and Y coordinates, via a mapping function, to KRLPos' X and Y. The new KRLPos object is then put in a ConcurrentHashMap using the symbol id as key. The ConcurrentHashMap holds each block and its' KRLPos object.

When reacTIVision recognizes that a block has been moved, the updateTuioObject function is called, which simply updates the KRLPos of the moved block identified by symbol id.

If a block is no longer recognized in camera-view, the removeTuioObject function is called and then the object is deleted from the map. One important thing to note with our current implementation is the use of symbol id, i.e. the id of the fiducial marker, as opposed to the session id, i.e. the incremental id sorted after when the block was recognized, makes it possible to insert the removed block at the same place again at a later time, without breaking the toolpath. This means that three blocks in camera view with symbol id 2, 7 and 10 will have session id (if added to camera view in same order), 0, 1 and 2 – removing 0 and adding it again would change its' session id to 3. There are advantages and limitations to each approach; we have chosen to use symbol id, as it makes it easier for a user to see, the sequence of motion, by utilizing a constant id for each block, which also is visualised on the physical block.

As an unforeseen side-effect of the current implementation, is that placing just one block on the surface, lets the user control the robot in real-time by moving the block around.

The refresh method is called after each TUIO message bundle, and this lets us call our pathGeneratingBlocksAlternativeMethod, which is a method for generating a path for the robot to execute. This method starts in a new, separate thread, if one does not already exist and checks the hashmap of current recognized objects is being tracked and if a connection to the KR C4 controller has been established. The reason for executing this part of the code in a separate thread, is to prevent blocking of the call-back methods presented before. If code ran sequentially, it would not register updates to blocks or new additions. If successful, a for-loop runs through all KRLPos

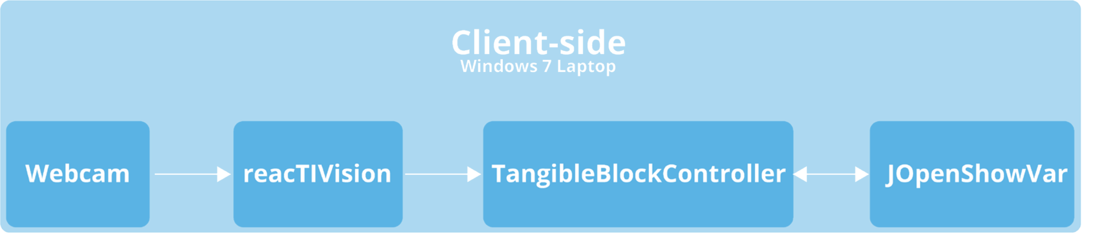


Figure 26: *An overview of our Tangible Blocks setup. The feed from the webcam is analysed in the reacTIVision software, which sends the data to our TangibleBlocksController.*

objects in the positionObjects map. Within each loop, it checks for the current position of the robot, reading the POS_ACT variable of the KR C4 controller and afterwards enters a while-loop: while the robot is not at the current position, it sends a KRLPos with the current position to the controller. This makes sure that the robot does not receive conflicting move-commands and that the robot does not move to another point, unless the intermediary point has been reached. Due to the use of approximation on the controller, a tolerance has been implemented in the position check. In our current implementation, the toolpath is repeated indefinitely, until no blocks is on the tracked surface or the robot is turned off; a pause block could easily be implemented, but the continuous feedback from the robot moving proved beneficial for adjusting the toolpath in real-time. An overview of the whole setup can be seen on Figure 26.

Input-Output Mapping

As the camera cannot view depth, like a Microsoft Kinect (Microsoft, 2015) can, only X, Y and angle of the marker can be inferred. This means that the X and Y is directly mapped to the robot based on assumption that the architect is working in front of the robot, as learned from our preliminary testing. A quick realization was that a one-to-one mapping between the interaction area, i.e. the surface area seen by the camera and the robot's area of movement made it easier for the user to recognize the result of blocks position; this made it possible to measure distances between blocks and place them accurately for creating a precise toolpath. Hence, the closer the physical appearance of the path created by the blocks, the easier it is to understand how the placement of blocks translates to robot movements. The more blocks used to create, e.g. a curve, the more precise control an architect have over the toolpath

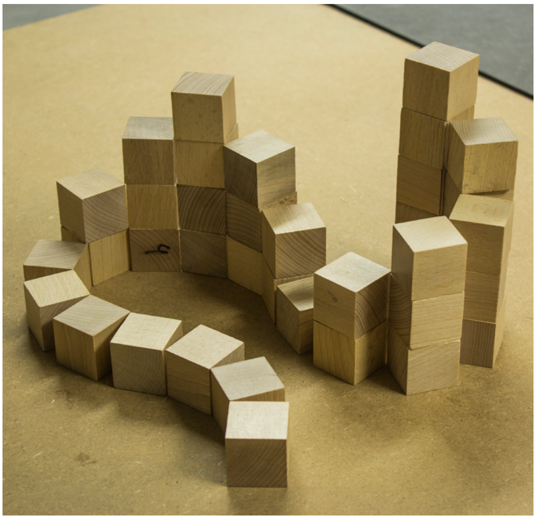


Figure 27: *During our experimentation with the Tangible User Interface, we brainstormed ways how blocks could be used to generate a toolpath for the robot to follow. Here a curved path, varying in height, has been made.*

– increasing the fidelity of the toolpath creation. In our initial implementation of the tangible blocks, only X and Y was used and Z was constant. Later experiments used the rotation of the fiducial marker for Z, i.e.

adjusting the height of the end-effector, however the coupling between rotation and height was not easily understood. Further testing showed that mapping the yaw of the end effector was easier to understand and was more appropriate for manipulating forms in sand, as height had to be constant regardless. Using a Kinect camera could build on this, adding height, i.e. building blocks on top of other blocks to fully visualize the toolpath in three-dimensions; Figure 27 illustrates this idea.

We limit the description to our current implementation of X, Y and yaw, using the blocks on a two dimensional interaction area. In our prototype, the blocks only symbolize waypoints for toolpath generation, but one could imagine that the blocks could have a variety of functions, such as four blocks creating the work area boundary dynamically.

Type of Interaction

The previous prototypes focused on mapping the robot end-effector to the movement of a hand, either directly or through an artefact like a mouse or keyboard. In this experiment, the interaction was done through the placement of blocks, which physically symbolizes waypoints that the user can move, remove or rearrange whilst the robot is executing the toolpath continuously. Using one block enables direct control of the robot, however, when using multiple blocks, the interaction shifts to somewhere between direct and semantic; the user defines the path of motion that the robot should follow in its' own work area. Whilst the robot is moving alone the defined path, the user can change the blocks positioning to reconfigure the robot's toolpath. However, using physical blocks has its' limitations when it comes to visualizing information about state; the only information easily inferred is the position and rotation, but it is somewhat limited by the physicality of the blocks, which could be thought of as a placeable pixel, i.e. the physical dimension limits the resolution of the toolpath. Even though a Kinect could be implemented to recognize depth, the blocks would have to be redesigned to be rotatable like a twelve-sided die.

Level of Autonomy

The robot implements the same safety behaviour as the previous prototypes, however, by interacting through the arrangement of blocks, the robot can assume new behaviours. As the toolpath, e.g. the path generated by the placement of blocks, is continuously known, the robot can modify the path or add movements to it. Our current implementation only chains the start point and end-point for repeatability. However, taking inspiration from graphic editors, such as Adobe Photoshop (Wikipedia, 2015) or Adobe Illustrator (Wikipedia, 2015), one could imagine how functions, such as mirror, scale or rotate, could be implemented for manipulating the toolpath. These functions could be used dynamically in a behaviour that builds upon the user's arranged path, e.g. making a curve with blocks, hereafter the robot takes the input and mirrors it, creating an ellipse – creating flawless, symmetric shape. This type of behaviour, as opposed to the previous example of shared control, is happening sequentially, taking the user's input and afterwards building upon that.

This third type of behaviour makes the control of the robot dynamic, as the toolpath altering behaviour can be viewed as a filter, for which the input is changed by the robot. However, for the architect to perceive the robot's intentions, this behaviour have to be visualised in some way. Looking ahead, the robot's knowledge of the toolpath, makes it possible to implement a rule-based system, i.e. as an aesthetic behaviour (Association for

Robots in Architecture, 2013, pp. 83-92). Math has long been a source for inspiration in art, it could be used for implementing aesthetic proportions, like Euclid's ratio, i.e. the Golden Section or implementing recursive algorithms that creates a toolpath based on user input (Gamwell, 2015).

Role of the Robot

The robot is essentially used as a tool for executing a sequence of movements, however, due to the supervisory behaviour and the possibility of input-altering behaviour, the robot takes on all three roles; a tool, a supervisor and when directly affecting the outcome of the user's input, it's a peer contributing to the generated form. We look more into the input-altering behaviour and the peer role in our upcoming sand exploration section.

Summary of the Explored Prototypes and Lessons Learned

We set out to evaluate a variety of interaction technologies in order to find common attributes that support collaboration between human and robot. These technologies were continuously evaluated and improved during the implementation. Through the experimentation with these technologies; unforeseen experiences were gained, such as how the shared work area should be designed and how this setup affects the physical relationship between the robot and architect. They share a common goal, which is to manipulate a material within a shared work area.

Our prototypes show how communication is mostly done from human to robot, and from robot, through manipulation of material, to the human – this will be further discussed in Chapter 8. By using the presented initial framework, we emphasized how the different technologies have advantages and limitations in each regard of the model, this will be further explained in detail in the subsequent chapter. We began by implementing traditional HCI input devices, the keyboard and mouse, seeing how these devices simplify the interaction, which limits the ground for collaboration.

Afterwards, we proceeded to examine Leap Motion, which is analogous to a mouse, but three-dimensional – the Leap Motion works great for doing manual manipulations, i.e. taking control of the movements. Our last experiment, Tangible Blocks, uses a different approach; instead of mapping something directly to the position of the end-effector, it was mapped to the path that the end-effector moves along. Finally, we see the robot taking on different roles in different ways, such as becoming a peer by sequentially altering input or by sharing direct control with the architect. Throughout the experiments, working with these prototypes sparks new ideas for future directions for each of the technologies; these future directions are summarized in general in Chapter 8, the future work section.

SAND EXPERIMENT: CURVE EXPLORATION IN A GRANULAR MATERIAL

In the following section, we conceptualize a design experiment of form exploration with the specific purpose of iteratively exploring a curve in sand. The technologies are presented in the same order as they were presented in the previous section. We examine how a user would explore a curve-like path in sand, using each of the previous presented technologies; the experiment is not meant to advance organic shapes in granular material, but instead allowing a first-hand view of the advantages and limitations of using the technologies listed to manipulate material for a fairly specific purpose. Even though the mouse and keyboard served as an introduction to interacting with a robot, and therefore limited in functionality, they have been included as an element for comparison to traditional HCI. In a design process, this would only be a small part of selecting and combining shapes, hence none of the developed forms were later cast in concrete. The overall experiment is used to give an account of these technologies, which will be used as a foundation for comparison in the subsequent chapter and how the individual interaction types can be further investigated in a great deal of directions in the future.

The Setup

In addition to the current setup, 40kg/24 litres of granite-based sand have been poured into a plastic storage box for cleaning and handling purposes; the experiment can quickly be reset as well. In addition, a 3D-printed spatula end-effector is used for creating various forms in the sand – see Figure 28. The spatula was printed before the box setup was considered, as it consequentially is relatively large for the box. Our focus is solely on a form of subtractive manipulation, i.e. clearing a path through the sand. The sand has been made slightly wet, in order to better retain the shape during manipulation.

Keyboard Interaction Prototype

When using the keyboard, the end-effector is constantly level with the surface of the sand, therefore the user can only move in the X, Y plane as described previously in the chapter. This means that the robot has to be jogged manually into a starting position within the sandbox. A curve can perhaps be recognized on Figure 29, however it required multiple passes, otherwise it would simply result in two orthogonal lines; one being

Figure 28: *The spatula, which was 3D printed on an EOS Formiga 3D printer (EOS, 2013), has been mounted on the mounting flange of the sixth axis of the robot. The image showcases our testing of the structural strength of the spatula in granite sand.*

Figure 29: *A top view of the sandbox after experimenting with a keyboard for controlling the robot. An error can be seen in the bottom, where the end-effector goes too far.*

thin and the other thick, due to the fact that the end-effector is fixed throughout the movement. One does not need to be a fine art connoisseur to recognize the results as a direct indicator of the limitations in this type of interaction. However, it presents one interesting aspect that constraints of an axis can create unexpected results, e.g. the end-effector being fixed creates a thin and wide stroke through-out box. Beside the fact that this type of interaction is quite limited, it gives the user a sense of directly controlling the robot and even as the spatula pushes through sand, one unconsciously thinks of the possibilities for creating form within the sandbox.

Mouse Interaction Prototype

The next traditional HCI device is the mouse, which is constrained to the X, Y plane as well. Like the keyboard, the robot has to be manually jogged to a starting point of the sandbox and the end-effector stays level and its' rotation is fixed.

When comparing Figure 30 with Figure 29, the difference of the curves is conspicuous, as the more organic lines become easier to create using a mouse. Using the mouse to form lines in sand is gives a relatively similar feeling of sketching or drawing, as it is a quick method of iteratively manipulating sand.

Leap Motion Interaction Prototype

Overcoming the limitation of a two dimensional plane of traditional HCI devices, we look towards the use of a Leap Motion controller to enable three dimensional interactions with rotation on each axis. First, we look at using X, Y, Z with yaw, pitch and roll constant. Using the Z-axis for height, consequently moving the fixed spatula in three dimensional space enables deep and

Figure 30: *A steep curve can be recognized in the above picture, which was done in only one movement by using a mouse. Notice the more distinct stroke difference in the curve.*

Figure 31: *By keeping yaw, pitch and roll constant throughout the motion, some of the same features as the previous experiment can be seen. However, as depth is added to the curve by varying height (Z-axis).*

shallow points on the curve, however as A, B, C is fixed the result resembles the previous ones – see Figure 31. Hence, we did another experiment with full control (X, Y, Z, A, B, C) of the end-effector, which allowed for more control over the end-result as seen on Figure 32.

However, as the complexity of the movement increased, the end-effector became more difficult to control, which might be because of the mapping between Leap Motion and robot; in order to have larger work area the input from the Leap Motion is up scaled, which makes small, unconscious movements translate into jittering of the end-effector. A delay is also present, due to the fact the robot has to move all the axis at the same time in order to have same direction and position as the interacting hand – increasing the speed decreases the delay, but makes it more unfriendly to control. By unfriendly, we mean that we see that the robot's speed can make the user not at ease, since small unconscious movements translate to the robot, hereafter the user will consciously try to correct this and causing the end-effector to jump a bit up and down (or from side to side). The small interaction area poses a real limitation of controlling the robot, when using a Leap Motion controller.

Tangible Blocks Interaction Prototype

Moving away from directly controlling the robot through in-air gestures, we look towards our implementation of a tangible user interface, using blocks to create a curve through sand. Our first experiment is based on the tangible blocks representing X, Y and Yaw; keeping the height constant.

Figure 32: *The above result is caused by the rotation of the interacting hand being mapped to yaw causing a straight stroke throughout the curve.*

As seen on Figure 33, the result is much different than the previous experiments, this is due to mostly two facts; one being that the robot repeats the toolpath, which connects the starting point and endpoint (see the top line of the triangle on Figure 33) and the second being the parametric aspect of the tangible blocks, as the authors played with rotating the blocks, affecting the yaw of the end-effector and thus deforming the lines.

In the next experiment, we let the robot take the role of a peer by giving it more control of the toolpath, and hence taking input from the user and creating a curve based on three blocks; one being the start point, auxiliary point and end point – approximating a curve based on these points. In our TangibleBlockController, a curveGeneratingBlocks method has been implemented so that it simply takes the coordinates of three blocks, C1, C2 and C3 respectively, and uses our coded CIRCProxy program on the KR C4 controller. This CIRCProxy program utilizes the CIRC motion command, which uses approximation and three parameters (start, auxiliary and end) for creating a curve. The curve can be seen on Figure 34.

Using the CIRC motion command (CIRC for circular motion), a half-circle can be created with just three blocks and a circle with just two blocks. The varying width of the curve is due to a constant orientation of the end-effector, this could be programmed differently using $CIRC_TYPE=#PATH (KUKA Robotics, 2010) to program a path-related orientation within our CIRCProxy program. This experiment can be categorized as direct manipulative – in the sense that you interact with physical objects that corresponds digital elements directly translated to motions of the robot. i.e. you place three blocks that corresponds three points in a coordinate system, where the system then calculates an arc across these points.

Overcoming the constraint of using the X, Y plane in our Tangible Blocks example and the limited interaction area of the Leap Motion, we now try to combine the two in order to facilitate a system that enables toolpath creation/manipulation and direct control during the execution of this.

Tangible Blocks + Leap Motion Interaction Prototype

As a last experiment, we draw on our experiences of implementing and using the above technologies to see how a combination of the two would help the architect in a similar kind of experiment. In this experiment, we use the Tangible Blocks without circular motion, i.e. point-to-point creation of a toolpath and see how a Leap Motion controller can influence the collaboration between human and robot. We start by implementing the Leap Motion as a way to continuously adjust the height of the current position of the end-effector, enabling X, Y and Z. We utilized the previously implemented grab gesture to control when the Leap Motion would set the height for the toolpath. As opposed to the three blocks from the earlier Tangible Block experiment, we employed a total of five points to create more of a curve than a triangle – See Figure 35. A short video of the combination of tangible blocks and Leap Motion interaction prototypes can be found at http://cr.coel.dk (Laursen & Pedersen, 2015).

As opposed to the previous experiments, where we prototyped the movement before placing the sandbox underneath, we could prototype the in-air movement and afterwards direct the end-effector into the sand using the Leap Motion controller and grab gesture. We experimented a lot with the

Figure 33: *This curved triangle was caused by the repeated toolpath movement of the robot, using blocks for controlling X, Y and yaw.*

Figure 34: *Using three blocks, an almost perfect curve has been created. This was done iteratively, adjusting the positioning of the blocks. The variable stroke is caused by the constant orientation of the spatula.*

Figure 35: *A curve can be deduced in the bottom of the picture with a connecting line in the top, which is due to the repetition of the movement. Through experimenting with parameters, we unexpectedly came up with* ***The Dented Curve.***

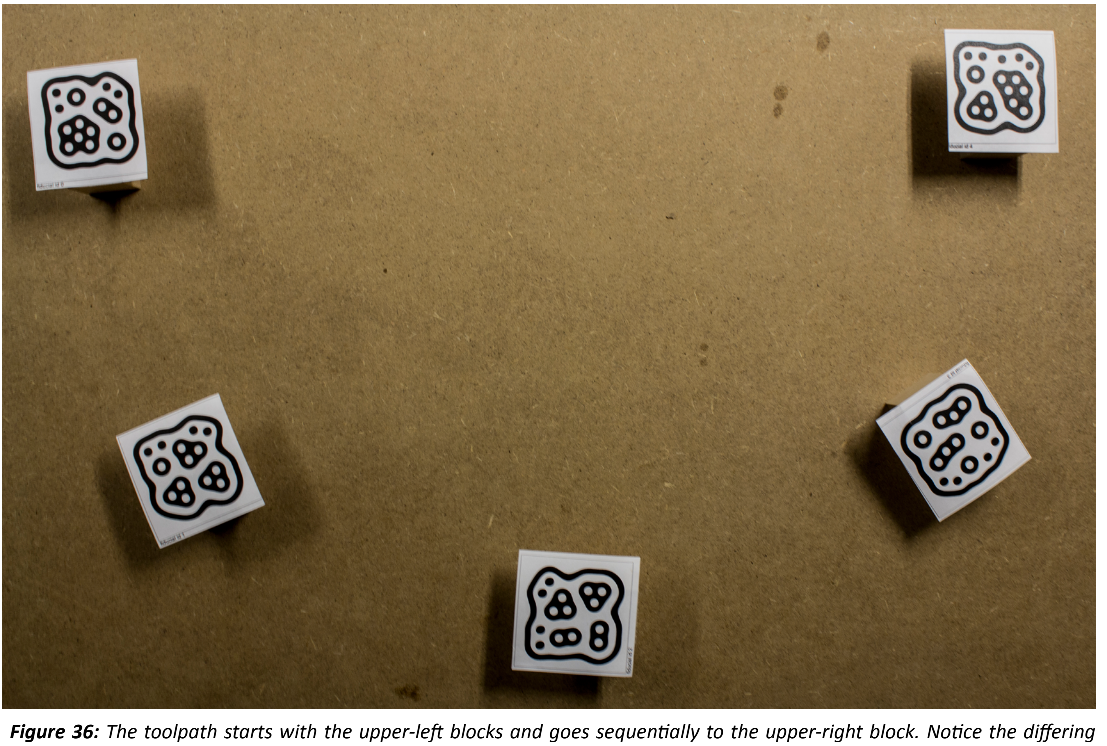


Figure 36: *The toolpath starts with the upper-left blocks and goes sequentially to the upper-right block. Notice the differing orientation of the markers, causing the end-effector to rotate during execution.*

slope of the curve using some of the blocks in between and rotated them to see how the rotation of the spatula would look like. The authors note that iteratively adjusting the movement of the robot, affecting the shape, made for a playful and enjoyable experience. On Figure 36, an image of the blocks that created the form can be seen.

This combination of technologies concludes our sand experiment, which shows us how the different interaction types affect how a curve can be made in sand. We saw how traditional HCI devices failed to enable a more parametric type of interaction and served simply as a quick and direct method of controlling the robot. The combination of the Leap Motion controller and tangible user interface made for a useful, enjoyable interface that allowed quick iterations and prototyping of movements; adjustments could be made on-to-go and allowed the exploring of different curves.

SUMMARY OF CHAPTER

In this chapter, we presented our preliminary prototype that have been used for exploring online and off-line programming of a robot, including how it can be directly controlled using a Leap Motion controller. Afterwards, we presented our setup and configuration of our testing platform, consisting of a solid work-table with a KUKA KR6 R700 sixx mounted on the table top and a KR C4 controller placed on a shelf underneath.

We used the key aspects identified in the previous chapters to sketch out an initial framework for robotics in architecture. We presented four different ways to either directly or indirectly control the robot, of which we used the initial framework for describing the implementation, input-output mapping and the evaluation of interaction type, level of autonomy and role of the robot.

This gave us an understanding of how different interaction types affected the overall collaboration between human and robot and what behaviours could create the foundation for collaboration. We hereafter used these technologies in a specific and simple form exploration experiment with sand to see how the interaction technologies behave. We built upon the experiences gained by using these technologies and combined the Leap Motion controller with our Tangible Blocks interface. We will in next chapter evaluate these prototypes compared to existing practice of parametric visual programming through the use of the KUKA|prc plugin for Grasshopper.


ROB | ARCH
Robotic Fabricatic in Architecti Art, and Design
mathematics + art
A CULTURAL HISTORY
LEAP
The Reflective Practitioner
How Professio
Think in Acti
Donald A

7

Evaluation of Protypes and Framework for Robotics in Architecture

In the following chapter, we evaluate the proposed framework by using it for analysing the developed prototypes, in order to sketch out the direction for future development and discuss the viability of the framework for collaborative robotics in architectural design. Furthermore, we discuss examples of how robots can contribute to the creative process of form exploration in order to support and assist the architect providing unexpected results.

COMPARING EXISTING PRACTICE WITH PROTOTYPES

We have explored different methods of interacting with an industrial robot in the creative process of form exploration. In the following section, we will elaborate on the strengths of each prototype, compared to the existing practice of visual programming (KUKA|prc). In the beginning of the thesis, we identified that the process of sketching out a design until implementing it by the use of a robot was long, tedious and slow-paced. In addition, we identified a discrepancy between, what you did digitally (input) and the output created by the robot in the physical world. Furthermore, the ease of use was of concern as well, as the average architect, as seen in earlier chapters, is normally lacking advanced programming skills. Consequently, the entry barrier should be lowered, to facilitate uncomplicated use of the robot; the learning curve should not be this high, in order for novices to get an understanding of how they can be used in the design process.

In the following section we will describe and elaborate on the existing process of using KUKA|prc in order to create a curve through three points and how one can experiment with the adjustments of the points' parameters.

Existing Practice of using KUKA|prc – a parametric design tool

In order to create a curve using the KUKA|prc, the most basic components have to be connected. In the initial phase, all work is done virtually in Rhino3D, Grasshopper and KUKA|prc. Among these basic components, is the KUKA Core, which is a main component that other components, such as a tool, robot and command components, link to. We use the KUKA KR6 R700 sixx robot component and the custom tool component, in which we enter the X, Y, Z offsets of our spatula tool. A 3D model of the tool could have been rendered for this component for simulation purposes, however it required too much time to be worth it for this description. In order to replay a simulation and do our own collision/

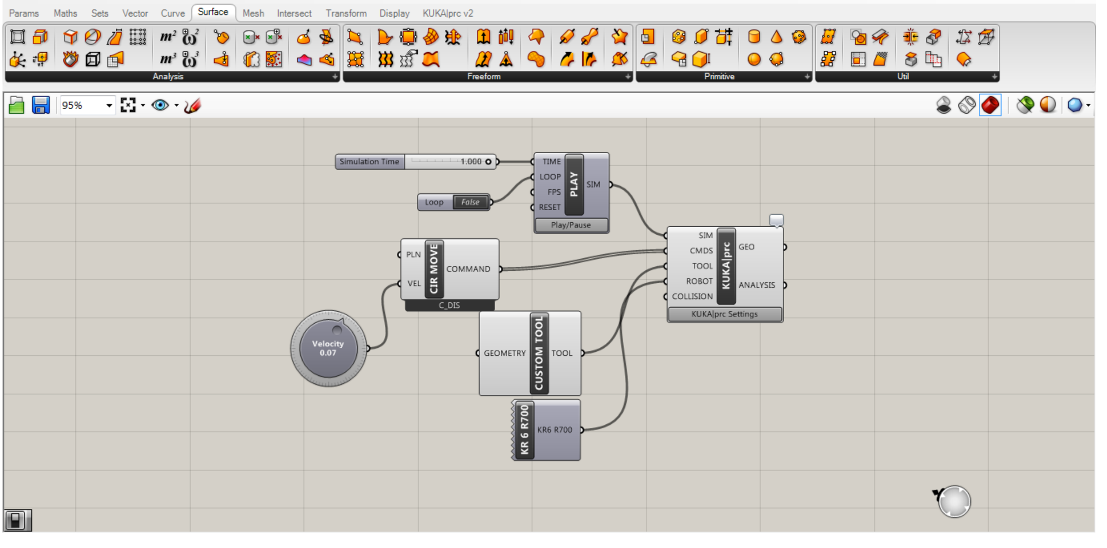


Figure 1: *The right most component is the KUKA Core component, to which all other components connect. This is the most basic setup required for creating a curve.*

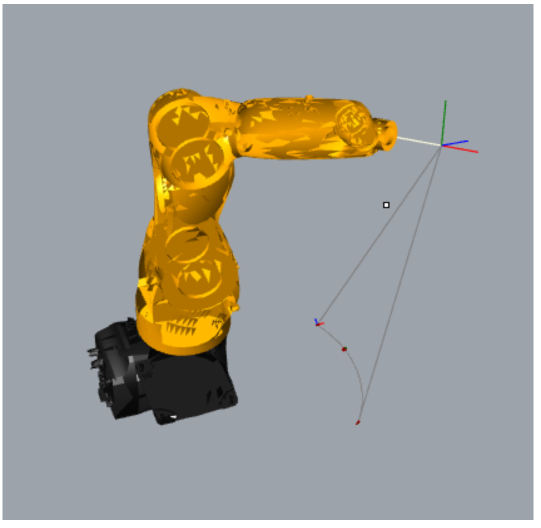


Figure 2: *The grey lines mark the tool path, whereas start position and end position is set to the home position of the robot. Adjusting the curve involves changing each of the points along the curve, which can be seen as small red dots.*

reachability testing, we add a KUKA Play component, where a numerical slider has been attached for advancing the simulation – see Figure 1. The CMDS input of the KUKA Play component takes either single commands or a list of commands, in our case, we have used a single CIR command component.

The CIR component takes two inputs: velocity and a list of planes, where three is needed to create a start, auxiliary and end point. The three planes have been defined within the component and placed in Rhino's 3D viewport. A knob has been added to the velocity in order to control the speed of the curve movement. The code in Figure 1 generates the path seen on Figure 2.

As we do not have a virtual replica of our work platform, we have to measure the height and position of the sandbox in order to arrange the curve correctly within KUKA|prc and test it out on the physical robot. This is done iteratively until we have positioned the end-effector correctly. The curve can be adjusted by re-positioning each of the place or rotating them. This process is much like the one we observed at Aarhus School of Architecture; the initial setup and configuration of the toolpath is time consuming, but afterwards it is faster to experiment with the curve, but still slightly time consuming. Every time a change has been made to the curve, the code has to be simulated and checked for reachability, afterwards it has to be compiled and transferred by USB drive to the KR C4 controller before testing. Luckily, collision and reachability testing is integrated in KUKA|prc as opposed to the observed architects use of HAL with RobotStudio for checking reachability and collision testing.

Strengths and Weaknesses of the Explored Interaction Technologies

In previous chapter we went through a minor design case in which we explored the strengths and weaknesses of each type of interaction, compared to each other. In this section, however, we will consider them individually and elaborate on the strengths and weaknesses they each have compared to the existing practice of KUKA|prc – a visual parametric programming environment, developed specifically for the KUKA robotic platform. We found this platform to be the most commonly used within the academic and industrial world, in addition to be relatively newly developed, while also being usable with our own KUKA robot. The following section is an overall assessment of strengths and weaknesses compared to existing practice. Hence, they will be discussed in conjunction with each other and not separated in "strengths" and "weaknesses".

Keyboard-based Interaction Prototype

One of the obvious *strengths* that are shared amongst all of the developed prototypes during the thesis, is the connection between input and output of the digital and physical world. While the keyboard has some obvious drawbacks regarding control of the Z-axis, the simple approach of being able to control and see that input digitally has an equivalent physical output in X and Y axes in real-time, makes the interaction meaningful. The connection between the digital world and physical world become more relatable to each other, when compared to existing practice. This in turn gives the user the ability to overcome another perspective which we identified during our empirical study – the need for a more effective iterative process, when exploring form. Thus the interaction type helps the architect to generate an output that is closely related to the input.

However, the overall ***weakness*** of the interaction type, is that it does not create the possibility of a lot of variations and consequently limits the current motions and trajectories of the robot. One could argue that adding additional controls on the keyboard could overcome this issue, e.g. by letting the PageUp and PageDown keys control the Z-axis. Thus, form exploration will be limited to very simplified forms and therefore limits the capabilities of creativity which of course is a major drawback.

Furthermore, the ***level of autonomy*** is rather low, as the robot is directly controlled on the basis of the architect's input on the keyboard, which in turn positions the robot as a tool. This could be subject for change though, e.g. if you change the robot to automatically smoothen the diagonal trajectory to a more circular movement which is seen in old 2D games. Thus, some of the control will be handed over to the robot which in turn changes the role of the robot to be either supervisory or a peer.

For future use, this method of interaction will clearly not outperform visual programming such as KUKA|prc as it does not possess the huge scope of complexities and does not allow the architect to experiment parametrically.

Mouse-based Interaction Prototype

The mouse has a long and profound history within traditional HCI and is still the go-to device when navigating the desktop of computer. Most people know how to use a mouse without any further introduction. Thus, a ***strength*** is that it makes it a rather intuitive and simple device for interaction, and it does not change considerably when working with a robot. However, it sets some obvious constraints, as it mainly operates with X and Y axes in a two-dimensional space. This in turn can cause confusion, when you approaching the problem, in regards to how the input is mapped as output. However, when using it, the connection between the input and output becomes clear, and just like the keyboard prototype, it brings the architect closer to identifying the relationship between your actions and the action of the robot. Furthermore, the nature of direct control lets the architect mock up motions and movement paths more quickly, in order to rapidly explore forms and shapes in sand.

Weaknesses of the mouse is the limited range of motions, which one is able to create with the mouse. It could be argued that by integrating the mouse wheel as an input type to change the Z-parameter, more elaborate explorations could be carried out. In addition, orientation of the TCP could be directed through the use of the right mouse button just like in a CAD program, e.g. Rhino, when you are panning around in the perspective viewpoint (3D viewport). Additionally, the complexity of the generated forms and shapes are extremely limited, compared to the existing visual programming software. In fact, this will always be a problem, when it comes to real-time management and programming of robots in the context of architectural design.

The ***level of autonomy*** is low, since the dimension of the interaction is of direct manipulation. This in turn indicates that the collaborative dimension between human and robot is low, i.e. the role of the robot remains a tool for design. However, in the case of safety we see, as indicated in the previous chapter, that the robot takes on a supervisor role, ensuring that the architect cannot do any exaggerated motion that violates the workspace boundaries, by taking over control on affected axis.

Leap Motion-based Interaction Prototype

By moving to a gesture-based interface, we adapt some properties of communicating intention as we do it in human-human interaction. Rich information can, as we elaborated in Chapter 4, be communicated through communicative and manipulative gestures. In the case of using the Leap Motion interface, the communication of intention corresponds well with, what you intend to do, and the reaction produced by the robot. Which, when compared to a purely syntax and component-driven digital approach like the KUKA|prc, is less obvious. In addition, you get a very versatile way of interacting with the robot. You can by direct manipulation control the robot in real-time, as you try out different forms. But you are also able to control it through a semantic dimension, where you through a series of gestures symbolise a curve or an arch, and indicating when and where the height (Z-axis) should change in non-real time. In this way, the robot only acts, when the architect has finished the series of movements, which it has to replicate.

Again, the time it takes from an idea until it is being performed by the robot, takes only a small amount of time. In addition, it is possible to re-iterate movements continuously without timely breakdowns in the workflow and interaction. New aspects, or dimensions that emerge, compared to existing practice, through this type of interaction, is collaboration between human and robot. The architect has the possibility to let the robot control the X and Y movements, while the architect controls the Z-axis and orientation of the tool attached. Thus, the ***level of autonomy*** reaches a dynamic level, where both human and robot takes part of the control and participate in the creative process and the stage of form exploration.

Tangible Blocks Interaction Prototype

Depending on how the blocks are used, the interaction type changes between the direct and semantic dimension. This was also illustrated in the walkthrough of the prototypes in previous chapter. By making the generation of forms more tangible, i.e. by using block elements for creating waypoints in physical space as a physical representation of the digital inputs, the interaction becomes more meaningful compared to the KUKA|prc method. This also builds upon the same modality of KUKA|prc, where the architect inserts waypoints into a virtual 3D space. It makes more sense to create an outline of a curve and then changing the radius of the curvature through physical blocks than to do this in a virtual environment, since you can visually understand and comprehend the changes made directly. In this way the placement of physical blocks as digital waypoints for the robot to move along, connects the digital world to the physical world better than existing practice. This in turn makes the possibility of re-iterating faster and the process of adjusting parameters more simple. The continually evolving ideas and designs are better facilitated through this dimension of interaction. The scope of complexity and diversity of robotic motions are however still better handled in current practice, however we believe that adjusting the design of the blocks and using a 3D camera can overcome some of these limitations.

The *level of autonomy* can also vary between direct and dynamic, and the role of the robot changes accordingly throughout the interaction and use of the robot. It has the ability to take on all roles, which in conjunction with the changing level of autonomy, are dimensions the KUKA|prc simply cannot offer the user in the current state.

Tangible + Leap Motion Interaction prototype

Basically the combination of the two adds a range of possibilities to the way an architect would create and explore forms in, e.g. granular materials. The blocks act as the way to create the baseline of the curve, whereas the in-air gestures acts as a way to change the parameters of the curvature, e.g. adjusting the height of the toolpath generated by the tangible blocks. Furthermore, the orientation of the tool can be managed real-time as well to continuously adjust the creation of the form. As most strengths and weaknesses already have been accounted for in the above sections of the individual prototypes, we do not repeat them here. The obvious strengths lie within the fact, that the combination of the Tangible Blocks and Leap Motion prototypes sketch out a direction for where future research can be focused by solving the identified problems in current practice, such as a faster iterative design process, lower entry barrier and closer connection of the digital and physical world.

EVALUATION OF THE INITIAL FRAMEWORK

We now turn towards the initial proposed framework and how it was used as both a generative tool for brainstorming new interaction methods, as well as an analytical tool for looking at human-robot interaction in a collaborative design process. In our evaluation of the usability of the framework, we look at the implementation and experimentation of our previously explored prototypes from the last chapter.

Usability of the Framework

In order to evaluate the usability of our initial framework and later enter into a discussion about it, we look at our framework through two perspectives: one being the use of the framework as a source of inspiration for the further development of human-robot collaboration in the field of architecture, the second is using the framework for analysing existing interaction and collaboration practice within the field.

The Framework as a Generative Tool

Before diving into the implementation of prototypes in the previous chapter, we used the identified key aspects and dimensions of the framework to brainstorm different ways of controlling the robot. One of the most dominant dimensions of the framework is the interaction type: direct or semantic. The direct type of interaction lets the architect directly control the robot through gesturing, which can be done through an artefact or device, such as a mouse. Through the manipulation, i.e. movement or rotation of physical entities, the architect communicates his intention. A semantic type of interaction can be seen in the both the gestural and tangible user interface. The semantic is concerned with the symbolism that is related to your actions or gestures, e.g. illustrating a curve through in-air gestures or using the grab gesture to activate tracking. Using the type of interaction dimensions, we came up with the previously explored prototypes.

During our implementation of these, we relied on the framework as an inspiration for how these technologies could be implemented and especially how the robot could take control dynamically, raising its' level of autonomy and taking on new roles in this ongoing partnership. With the autonomy dimension, we uncovered several limitations in the different interaction types, e.g. how directly controlling the robot minimizes the possibility for autonomous behaviour based on user-input.

Based on the behaviours of the different explored prototypes, we saw, how the level of autonomy correlated with the roles that the robot can assume. This made the use of the framework more practical, as we, throughout the exploration started to use these roles to implement and categorise different autonomous behaviours. Different behaviours affect different aspects of the workflow, e.g. the safety behaviour is invisible to the user, until the robot hits one of the boundaries. The tool role means that no behaviour has been implemented and consequently, is the most basic implementation of the different technologies. On the completely opposite end of the spectrum the peer role describes, how human input is combined with robot behaviour to affect the manipulation of a material. An example of the peer role is how the robot uses the input of the tangible blocks interface and creates a half circle based on only three blocks. These three blocks can effortlessly be moved about to adjust the slope or radius of the half circle.

The Framework as an Analytical Tool

Using the proposed framework as an analytical tool, it can identify and guide an understanding of current products or prototypes in a context of architectural design with a focus on interaction and collaboration. Hence, we look at our proposed framework to see whether it can guide an understanding of the interactive and collaborative nature of the developed prototypes and how it might guide future work, in which the more complex prototypes can be developed. We will map the individual prototypes to the dimensions of the framework and elaborate on why this particular mapping in question takes place. However, a more elaborate analysis will only be done of the Leap Motion and Tangible Blocks prototypes, as we exclude the traditional HCI devices.

Starting from the very top of the framework, we first identify the given context in which the prototype is present. In the case of the thesis, the application domain does not change, thus it will be fixed on form exploration for all prototypes. However, this does not indicate that the framework is only applicable in the context of form exploration, we have merely chosen to only develop prototypes within this context, as it was where our observations took place and thus gave us the best starting point. Evaluating in the framework in another context might yield different results, because other interfaces might be more capable in e.g. fabrication of architectural designs. We map our two experiments according to the framework and discuss each dimension in relation to the particular prototype.

Leap Motion Prototype

Our in-air gestural prototype, based on the Leap Motion controller, gives us a direct way of controlling the robot by correlating the position of the hand to the position of the end-effector. But the prototype can also be used for semantic interaction, such as gesturing with a grab that symbolizes that the user grabs hold of the robot and so becomes able to move it to any position for afterwards to let it go by the user opening his hand; a sort of robot puppeteer.

Given the implementation of behaviours in our prototype, the level of autonomy is placed somewhere between direct and dynamic control, as the robot does not take full control of itself, unless the user tries to move the robot out of bounds. The robot can also take control of other axes, simplifying the interaction, but maintain full output possibilities, i.e. movement and rotation on all axes. This is exemplified by having the user control X and Y, but letting the robot control the Z axis, or vice versa; the result becomes a combination of both the robot and the user's effort and control.

The ***role of the robot*** is tied to the ***level of autonomy***, as more directly controlling autonomy affects the type of role that the robotic agent can enrol in. As we described in the previous chapter, a robot can have different ways of supporting and collaborating with an architect; we see how the robot can take a passive role as a supervisor, simply overseeing the work and taking control, when it is needed such as safety violations, which is opposite to current robot use, where a human assumes the supervisor role. The robot allows the architect to focus on the task at hand and makes his work easier by supporting him through various measures. This support is due to the shared goal of the two partners, as the robot can focus both on the task like the architect, e.g. manipulating form, and focus on providing guidance to the architect, helping him achieve this shared goal. Furthermore, we see that the robot can be of help in other ways in the future as well, perhaps even switch the roles of the two.

Tangible Blocks Prototype

Our tangible interface prototype gives us a semantic way of controlling the robot; by controlling its path of movement physically. The blocks represent waypoints or positions, which the robot follows sequentially. Due to the nature of our implementation, however if only one block is present, the correlation between block and position of end-effector is happening in real-time: moving the block causes the robot to move immediately. The blocks are limited to X, Y and yaw as opposed to the in-air gestural prototype.

As with the in-air gestural prototype, the control between robot and human is somewhat dynamic; the robot takes control if it sees that the architect is doing something that does not work towards the goal of creating form, e.g. moving away from the material to be manipulated. We also see how it can work together with the human in simple ways. An example of this is interpreting three blocks for the creation of a curve that can be easily adjusted. Thus, the robot takes on a peer role, where control is divided equally and done in real-time as opposed to the interventionary nature of the supervisor role and out-of-bounds behaviour. The robotic agent takes on multiple roles, either taking direct control together with the architect or more passively monitors the architects work with the robot as a tool, taking on a supervisor role and intervening when necessary.

Regarding the use of the framework as an analytical tool, we have mapped each prototype roughly to the framework and how each prototype correlate to the dimensions of the framework. As seen in Figure 3, each prototype is visualized as a colour positioned in each dimension. The positioning of the prototype in each dimension is relative and not an accurate depiction of where on a scale they are. It merely serves to

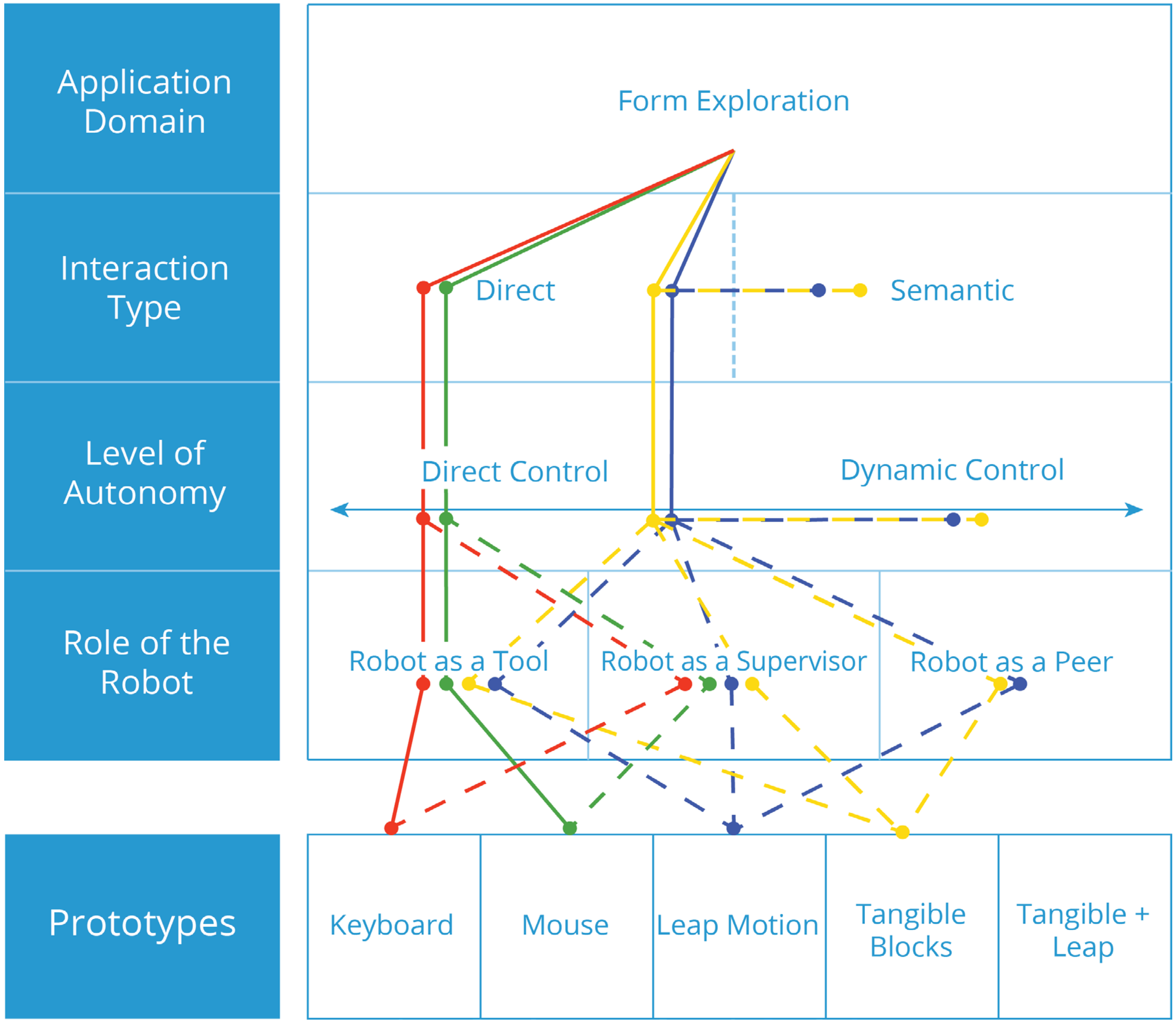


Figure 3: *Each prototype have been relatively positioned in the framework in regards to how they can be analyzed and described. The dashed lines indicate that in certain situations, the prototype in question can change its' position in the respective dimension.*

give an understanding and overview of the framework as an analytical tool regarding the presented prototypes.

Furthermore, the combinatory prototype of the leap motion and tangible blocks are not directly visualized on its' own, but they are the sum of the two individual positioned visualizations.

Summary of Framework Evaluation

We use the framework as a way to observe how the robotic agent can collaborate with a human, how human communicates intention through interaction and how they both support each other through various methods, such as curve interpretation, to reach an end-goal. In both the Leap Motion prototype and the Tangible Blocks prototype, the robot assumes the role of a peer, this is however due to different methods of sharing control, sequentially or parallel. The framework has been suitable for the development until now, we discuss the framework and its' limitations in the next section.

Discussion of the initial Framework

In the following sections, we will discuss the dimensions of the framework, the shortcomings regarding the domain and the framework itself and finally, how HRC, HRI and creativity is represented in the framework.

It might seem a bit trivial that the application domain only has one dimension in the framework, i.e. form exploration. However, by adding this dimension, we acknowledge that this framework is based on the activity of form exploration, but that other activities, such as robotic fabrication, might use the dimensions in a different order or even exclude some of the dimensions.

Currently, some previously mentioned aspects are implicit in the framework, such as learnability being an aspect of the interaction type dimension and the robot as a creative entity is implicit in the level of autonomy and role of the robot dimensions. For the sake of simplicity, the framework has these implicit aspects, but some of them could be expanded upon in the future, allowing for a more dedicated creativity dimension building upon our model from Chapter 4; specifically, which stages a robot can be a collaborative partner. We also see that both the in-air gestural and tangible interface implementations can be categorised as both direct and semantic interaction, but this is largely due to our implementation of the two in a combined manner instead of two separate implementations. We started implementing a direct way of interacting with the robot, but soon realised that the addition of a semantic gestural language could be helpful to signal certain intentions to the robot.

In future development, measuring physical states, such as heartbeat ratio etc., could be implemented as a way of inferring internal states and intentions from the human, e.g. determining if they are comfortable with the current state of the situation and so on. In addition, haptic technologies are a new an emerging technology used in robotics, as they can be used to measure torque and angle placement in collaborative tasks, where a human and robot are handling an object at the same time. This could also help the the architect feel the physical attributes of the material that is being manipulated. This could be added to our prototypes using the TECHTILE toolkit (Minamizawa, et al., 2012), which the authors experienced during a visit to the Tachi Lab in Tokyo. This could be described in the current presented dimension of interaction type, as a direct way of interacting with a dynamic autonomous level, as the robot will have to make assumptions on why the torque or angles are being changed by the human. In conclusion to this, we see that the framework is somewhat limited in regards to the current use context of form exploration and interaction, but it still remains as an inspirational tool for future development, and not a finite tool dictating a direction.

Currently, the dimension of autonomy goes from direct control to dynamic control, but we envision that the robot can be most autonomous in the future, leaving the architect to support and guide the robot in form exploration. This direction could involve a one-architect-to-many-robots relationship, as one architect could explore multiple form variations at the same time. This would put dynamic control at the centre of the dimension and then an addition on the right side: a fully autonomous robot involved in form exploration by simply creating variations of form and letting the architect give his opinion throughout the process. However, envisioning full autonomy might challenge the perception of how big a factor robotics should be in architectural design. As indicated in the former study, traditional chefs saw it as a challenge to the very traditional craftsmanship and suggested a move for a reduction in labour. This could be applied to architecture as well as it is a very traditional profession, at least it is a factor to keep in mind for future reference, when engaging in a co-design process as we have done.

Our framework only uses three roles from Scholtz (Scholtz, 2003), however we could extend our framework to encompass more roles, which could be interesting, when using the framework as a generative tool. An idea of another role could be the mentor role, which was added by Goodrich and Schultz (2003), i.e. letting the robot function as a scaffolding tool, where the possibilities and control are limited at first, but as the architect learns to interact with the robot, more functionality is emancipated and the robot gives the architect more control; basically teaching the architect more and more. Which, in HCI terms, can be described as a form of ***legitimate peripheral participation*** (LPP) (Bryant, et al., 2005), where the novice is only introduced to very basic features or tasks, and as they learn, more advanced and complete features or tasks are introduced. Sort of a master and apprentice relationship between robot and human. Whereas this terminology is often used regarding a community of practice, the use of robotics in architecture are an emerging community of practice, which are getting more common, as we have presented in the thesis. This mentor role could support novices in the use of robotics along with improving their skills.


MX3D

8

Discussion, Conclusion and Future Work

DISCUSSION OF EXPLORATORY DESIGN PROCESS

We will briefly summarise our design process and afterwards reflect, through a discussion, on the various stages that we have been through and finally, which steps lie ahead. We used a research-through-design approach in order to implement prototype interfaces for first-hand evaluation and as artefacts that we can gain knowledge from. Our initial grounding is in the form of a preliminary empirical study, along with a theoretical foundation that shaped our research questions. Our preliminary prototype is built on top of this foundation, and was used and evaluated to gain an initial insight into, how humans and robots can interact – this sparked some an unforeseen insight about the physicality of the robot, which was incorporated in the next prototype.

The initial empirical study was centred around the observation and interview of two students, one Master and one Ph.D. We limited our empirical study to the research field and therefore did not observe robots in architecture in the private sector. We could have seen compared the two sectors, focusing on the Danish firm Odico (Odico Aps, 2015), led by architectural researcher Asbjørn Søndergaard, formerly visiting researcher at Gramazio and Kohler, ETH Zurich. Additionally, our observation was limited to form exploration in granular materials, however, form exploration can be done in a multitude of other materials, which will have different requirements to end-effectors and robot movement. We realise that using one material as a constant to conduct different investigations into different interaction types has been helpful, but also realise that testing these interaction types to the manipulation of other kinds of materials will be essential in a more elaborate evaluation of the framework. We base our understanding of the domain on the related published research and interviews with leading researchers within the domain, Michael Knauss, Johannes Braumann and Sigrid Brell-Cokcan. Our preliminary robot was constructed based on an open-source blueprint, which was modelled after a slightly different robot than our KUKA KR6 R700 sixx robot – however, the lessons learned were universal for any type of articulated arm. The prototype helped inform how the KUKA robot should be setup in a collaborative setting for form exploration, but also informed how interaction should be implemented and mapped to the robot movements.

In Chapter 4 we presented and discussed theories and concepts that we intended to form a basis for our further investigation. These theories, in conjunction with our empirical studies, resulted in the proposal of an initial framework that we could use for inspiration for the development of the prototypes used for investigating the domain of robotics in architecture. This framework was based on key HRC and HRI aspects, which was summarised to the following: level of autonomy, interaction type, role of the robot, learnability and application domain. These aspects, or dimensions as we call them, were further reduced to form the framework as seen in Chapter 6: application domain, interaction type, level of autonomy and role of the robot. These dimensions aim to offer a tool for discussion of aspects that we saw fit the development of prototypes.

In late November, 2015, we received the KUKA robot for testing, which was up and running start December, 2015. Setting up the larger KUKA robot gave us an additional insight about the physicality of the robot. A desktop-sized articulated arm requires a great deal of space due to the circular nature of its' work envelope. Keeping the robot within the confines of the table top required a large surface and a stable construction to prevent table movement during robot operation. We also need to emphasize that the maximum speed of an industrial robot is not suitable for a close collaboration between human and robot. A human would not be able to react swiftly enough, if he had to get out of the way or prevent the robot from colliding with the table. We also realised that the high speed could startle users, as the robot makes no effort to let the surroundings know that it is about to move and at what velocity – this has happened on multiple occasions during testing. The physicality, by which we mean physical construction and movement characteristics, such as acceleration and speed, has to be considered, when designing for a collaborative relationship. One would not scare people walking down the street by making sudden movements towards them and noises. This hurts the very nature of a collaborative partnership, trusting the robot and its' intentions are of prime concern. We also quickly realised that sometimes a user will, through interaction, control the robot to its' outer positional extremities or collide with the environment at some point. Consequently, the robot cannot always rely on the human and therefore needs to implement self-supporting behaviour for the sake of its' own and in turn, the architect.

We saw two ways a robot could take control of the movements, one being to prevent errors and supporting the intentions of the architect and the second being the sharing of control in order to have both human and robot contribute equally. In the first behaviour, the robot simply assumes a supportive role, helping novices understand how their gestures translate into movements without the possibilities of colliding with anything. The second behaviour lets the robot assume a peer role, control is divided equally or sequentially and the product of this collaboration is an accomplishment shared by both partners. Creativity is the product of the two interacting and manipulating a material collaboratively. We see the progression from human control to dynamic control, and in the future, perhaps a fully autonomous robot acting based on human critique.

We started by experimenting with direct and traditional forms of control, the mouse and keyboard, which turned out to have some limitations in its' use, being a very simple source of input. We see that a disconnect between HCI and HRI exists, which is mostly due to the physicality of the robot and its' movements - positioning a tool end-effector in a three dimensional space, essentially. Translating mouse or keyboard input into movement on a two-dimensional plane has distinct limitations, when the complexity of the movements contributes to the complexity of the forms created. The different interaction types resemble different activities which architects often take part in, such as the in-air gestural prototype that felt a lot like sketching, by flicking one's wrist while moving the forearm one could create unique curves. This meant that the interaction types have a low entry barrier for architects, but this also has to be evaluated in a future workshop with architects, both novices and experts. The low-entry barrier is also due to the physical nature of the interaction, both input and output are placed within the physical world. In the end, we combine the in-air gestures and tangible interface to overcome some of the practical limitations, such as full three-dimensional control of the end-effector and are also overcoming the complexity of input in the in-air gestural prototype by transferring the responsibility of toolpath creation to the tangible interface.

We use these explorations to take a second, practically based look at our framework, using it as an analytical framework for describing the features and possibilities of our prototypes and inspiring new autonomous behaviour and roles for the robot. In the end, the implementations are somewhat limited by the fact that we only had the robot from November, setup and running in start December. This was due to the difficulty of renting one from either a research institution or manufacturer, as we have contacted most of the companies with an office in Europe. It could have been possible to rent the ABB robots at Aarhus School of Architecture for a limited time, however we would not be able to do low-level interfacing with them, due to a relatively restricted equipment loan agreement. Consequently more time would have allowed for more elaborate behaviours and implementations. These behaviours and improvements are described in the last section of the chapter.

In the end, we summarise our findings and discuss the framework according to the implementation and exploration of our prototypes.

DISCUSSION OF THE COLLABORATIVE AND CREATIVE ASPECT

Now we turn towards the discussion that builds upon our account on creativity, HRC and HRI in Chapter 4. In the following section we discuss our prototypes and framework in relation to the fields of HRC and HRI, including a discussion of how creativity is manifested in our experiments by relating them to our adapted model of the creative process presented in Chapter 4.

It is clear that the theoretical foundation of communication of intention stems from research into humanoid, as the scope of communication channels exceeds that of robots of the type industrial articulated arm. Therefore, as the industrial robot are non-humanoid and lacks the means of communicating through obvious simple and direct gestures, the two-way communication, and consequently collaborative aspect remains somewhat untraditional in the developed prototypes. However, as the architect commits to an activity involving the robot, the collaborative nature emerges in the way the architect perceives, how the robot intervenes in the process of form exploration. When sharing aspects of the control, the architect realises that he is not in total control and has to rely on the robot as an active reliable partner with the same intentions - the same applies to human-human collaboration, where we rely on each other and through interacting, reveal each others possibilities of action.

However, we do see limitations in the communication, and our prototypes does not entirely support collaboration per se. If the architect does not understand or perceive the robot as an agent, providing inputs of its' own to the creative process, the collaborative aspect is not present as it ultimately comes down to how the mental model of the architect is formed throughout the interaction stages with the robot.

The collaborative aspect that emerges in the various prototypes developed, rests in how the control of the work in question, is being shared between human and robot. In addition to this, also how the human perceives whether the robot adds something to creative process by interpreting the input provided and generating an output equal to, or similar to, the input. An example of this, is how the architect in the tangible blocks prototype delivers three waypoints as input, which the robot interprets as three waypoints for a curve. In this case, the robot makes a suggestion and communicates through the environmental channel, through the granular material, which the architect can choose to accept or disregard.

Concluding that the anthropomorphizing of the robot is not needed in order to generate meaningful communication between human and robot. The communication can emerge through the objects manipulated in the work area. We realise the potential of small, anthropomorphic gestures, such as shaking one's finger to signal "no" or "I can't do that" for future investigations; giving the robot more personality in relation to its' role, through new behaviours. However, even this could be replaced by visual projection onto the work area, using visual cues.

Looking at the activity of form exploration, the robot adds a touch of something unknown, which the architect cannot control himself. Taking a look at the creative process outlined in Chapter 4, we re-introduce our adapted model of the creative process, focused on form exploration. We argue that the creative process benefits greatly from this collaboration between robot and human, as the robot contributes to the architect's creativity through differing roles and underlying autonomous behaviour.

In current practise, robots are mostly used to externalise virtual toolpaths or structures through the use of various end-effectors. The generation of shapes and form is done virtually through the use of parametric design tools. Exploration is mostly done in two phases in current practise, virtually, i.e. simulating robot movement and in the physical world,

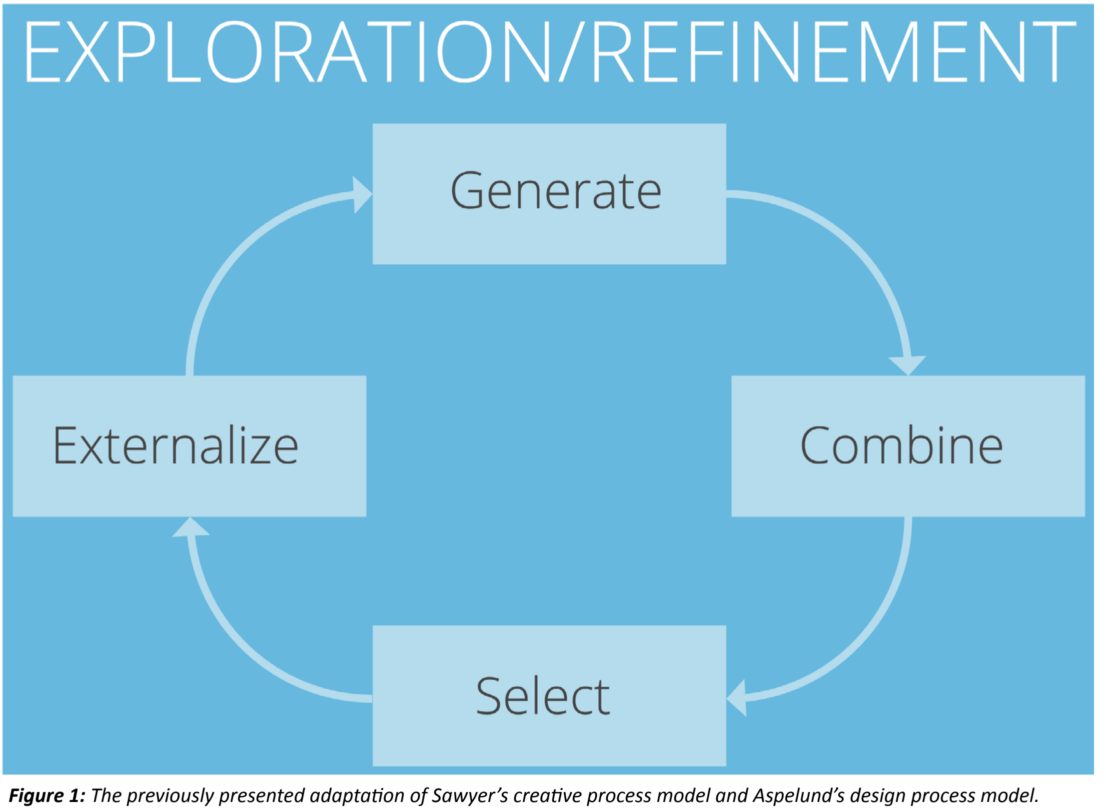


Figure 1: *The previously presented adaptation of Sawyer's creative process model and Aspelund's design process model.*

acknowledging the unforeseen material attributes during manipulation.

In our exploratory prototypes, we focus on iteratively experimenting with shapes and form in the physical world, letting the robot control more of this process than before. By collaboration and sharing control, the architect and robot share responsibility of generating new forms, combining, selecting and externalising them. The dimension of control lets the robot affect the form by combing its' own behaviour with the communicated intentions of the architect, therefore apply two input sources to affect a single output. The select phase is more or less unnecessary due to the iterative and continuous nature of the robot's movements, every change to the toolpath is being externalised. The explored prototypes help inform, how collaboration between robotic agent and human can result in creative outcome.

CONCLUSION OF THE THESIS

Through a research-through-design approach, we investigated how a robot and an architect can collaborate in a creative process, focusing on exploring form. By looking at existing practise in the field of robotics in architecture, human-robot collaboration and interaction, we grounded our explorations on empirical studies and relevant literature, which in regards to collaboration mostly has mostly been based on humanoid robots. Through the development and evaluation of the exploratory prototypes, we saw different ways in which a robot and human team could collaborate and how this collaboration could enrich the creative process of form exploration in architecture. We turn towards our previously introduced research questions that have guided our process:

- ***How can we design the interaction between human and robot, with the objective of improving the workflow of the architect's creative process?***
- ***How can the disconnect between the physical and digital world be reduced, when exploring form in granular materials?***
- ***What roles can a robot take in the activity of form exploration and how do these affect the architect?***

Research questions

Through the preliminary empirical study, we identified some key concerns we were guided by throughout the thesis and the development of the prototypes. These concerns were formalized as the research questions (see above), which we sought to answer through the design of exploratory prototypes. Central to the development was gestural (semantic and direct) and tangible interaction with the robot. We identified that the existing practice caused some major shortcomings in regards to the interaction and learnability of robot use, which became a prime concern. We learned in our prototypes, that the use of gestural means of NUI combined with TUI seemed to increased both learnability and ease of use compared to existing practice. Furthermore, making the interaction tangible improved the workflow as the connection between the digital and physical world made more sense. Input and output were better connected which signified a step toward a WYSIWYG approach which can decrease the cognitive load when working with unfamiliar systems.

Additionally, we sketched out a framework that would provided guidance during the development of the exploratory prototypes, overcoming the concerns highlighted by our research questions. This framework consisted of four dimensions (***application domain,*** *interaction type, level of autonomy, role of the robot*) intended to shed light on key aspects, which we found to be important in order to support a collaborative and creative process in exploring forms in architectural design.

This framework helped us elaborate on how the robot could be understood as an agent providing input for the creative process and how this could affect the envisioned teamwork between human and robot. In conjunction to this, we identified the robot taking up the roles defined in the framework at varying point throughout the interaction. However, the roles could be extended to encompass additional roles presented in the literature review in Chapter 4, e.g. the mentor role where the robot would gradually reveal more functionality for the architect as interaction was learned and mastered.

As a final note we want to emphasise that the framework is only an initial step towards defining a direction for research and design of collaborative robotic systems for explorative creative processes. It serves as an analytical tool for existing products or prototypes or as a generative and inspirational tool for further development of prototypes for the domain of collaborative creativity in architectural design.

FUTURE DIRECTIONS FOR ROBOTS IN ARCHITECTURE

Looking forward, we realise that multiple directions could be very interesting to investigate, both implementation-wise, but also overall for robots in a creative process. Throughout the last two chapters, we have hinted at future directions and explained some next steps. In this section, we go through the general future directions, starting by talking about the future for our tangible and in-air gestural interface prototypes, afterwards looking at future potential of the framework and how it might be expanded upon and finally, a look into the future for human-robot collaboration within the field of robotics in architecture.

Looking at our last prototype, combining the in-air gestural interface and the tangible block interface, we see that these can be improved upon in multiple ways. One way is switching to a Kinect 3D camera to incorporate marker and hand tracking. Visualising the robot's toolpath on and between these blocks, this increases the visibility of the relation between block position and robot movement. Using a 3D camera also overcomes the Leap Motion controllers limited interaction space, which adds the possibility of using both arms to interact with the robot, which poses some interesting questions to be explored. We also see the possibility of full body tracking to advance the communication of intention. An approach could be adapted from the project visualised Figure 2 and 3, where the contours of a landscape is modified and projected onto the granular material (Reed, et al., 2014). Thus, enhancing the connection between the physical and digital.

Another next step would be to explore the peer role more, observing the robot taking control in a variety of ways and afterwards, create a workshop for expert and novice architects to evaluate the last prototype. This workshop will effectively test the learnability of the prototypes, but also the possibilities of how the prototype can enhance their design process and let them iterate quicker – exploring form together with a collaborative, robotic agent.

Looking at how communication is facilitated, we see that the robot communicates through movement. Aside the use of projection on the shared work area, other modalities for communication could be interesting to investigate, since the use of an industrial robot opposed to a humanoid robot, poses some unique challenges regarding the communication of intention from robot to human. This could be explored through sounds, such as voice, nonspeech audio, or representational earcons (Blattner, et al., 1989). These sounds could let the user know the intentions of the robot, e.g. "I will not move further" or "I will take control now".

Our current implemented ***behaviours*** let the robot take the roles of a tool, supervisor and finally, a peer. More behaviours could be implemented or current ones improved in our current prototypes in the near-future, such as installing a webcam above the robot's work envelope. Using this webcam and the X11 safety plug, the robot could see if a human enters its' work area, consequently slowing down movements or going into a resting position – allowing the architect to correct or change the material. We realise that looking at other roles for inspiration can be interesting. Moving away from Scholtz's roles and Goodrich and Schultz's extension, we also acknowledge which roles are something that human and robot can exchange, such as the robot being the individual to explore form, based on some aesthetic measures, using the humans input as a critic and supporter. The collaboration becomes more complex, as the roles shift during the creative process. Therefore, this dynamic relationship between human and robot could be interesting to explore. The framework has helped us understand and evaluate the explored prototypes through the identified dimensions from preceding research.

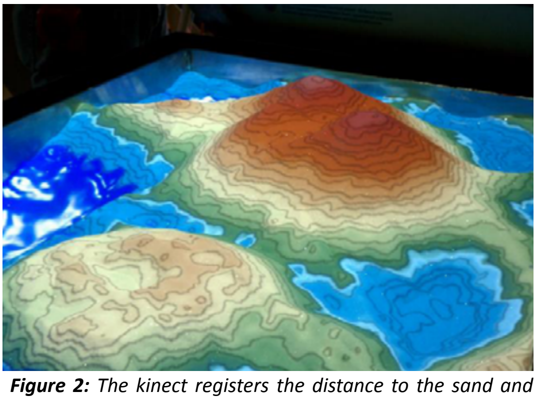


Figure 2: *The kinect registers the distance to the sand and the topographic map is projected onto the sand. This gives a picture of the elevantion profile and contour lines of a landscape. (Reed, et al., 2014)*

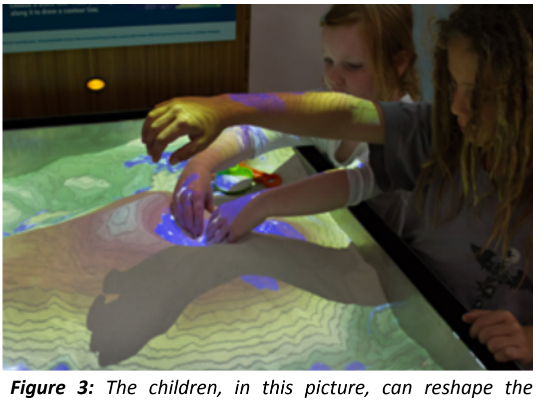


Figure 3: *The children, in this picture, can reshape the physical material and the projected digital representation of a landscape is simultaneously being changed. (Reed, et al., 2014)*

We acknowledge that the ***physicality*** of the robot is what sets HRI apart from HCI, but this direction could be very exciting to explore further - studying how architects perceive the robot as an active partner in their design process and how the characteristics of movement can invoke changes in this perception during a creative process. The physicality of the robot is also how the robot acts and reacts according to its' surroundings, looking at how the architect's interaction affects this, this poses an interesting question of how the robot should present itself and how this affects the collaboration.

For further evaluation of our ***framework***, a clear next step would be to apply it to other activities within the field, such as manipulating foam for sculpting arches and overhangs.

This could add another dimension, which relates either to the subtractive or additive nature of the end-effector. Different materials have different requirements to the robot's movements and in turn, the user's creation of a toolpath.

Our research has been confined to a single activity within the field of Robots in Architecture, but looking towards other professions' creative processes, such as industrial design, a robot might take other roles, e.g. an assistant, supporting and contributing in other ways, opening the design space of human-robot collaboration further. We also see that this collaborative relationship could be introduced in other contexts as well, including the fabrication of finished building parts, but also in the building process itself. This is the natural direction for future research, as the robot continues to become an integral part of our lives, supporting our actions and sharing our goals.

Bibliography

ABB, 2013. *IRB 360 - Industrial Robots (Robotics) - Industrial Robots - Robotics | ABB.* [Online]
Available at: http://new.abb.com/products/robotics/industrial-robots/irb-360
[Accessed 3 December 2015].

ABB, 2015. I*RB 120 - Industrial Robots (Robotics) - Industrial Robots - Robotics | ABB*. [Online]
Available at: http://new.abb.com/products/robotics/industrial-robots/irb-120
[Accessed 8 December 2015].

ABB, 2015. *RobotStudio.* [Online]
Available at: http://new.abb.com/products/robotics/robotstudio
[Accessed 21 December 2015].

ABB, 2015. *YuMi | ABB.* [Online]
Available at: http://new.abb.com/products/robotics/yumi
[Accessed 18 October 2015].

AGATA KYCIA, 2011. ___ *Digital fabrication /// 3-axis milling | agata kycia.* [Online]
Available at: http://agatakycia.com/2011/12/30/digital-fabrication-3-axis-milling/
[Accessed 25 December 2015].

Ambrose, R. O. et al., 2000. Robonaut: NASA's Space Humanoid. *Intelligent Systems and their Applications*, 15(4), pp. 57-63.

Araullo, J. & Potter, L. E., 2014. *Experiences Using Emerging Technology*. New York, ACM, pp. 523-526.

Arduino, 2015. *Arduino - ArduinoBoardUno*. [Online]
Available at: https://www.arduino.cc/en/Main/ArduinoBoardUno
[Accessed 20 December 2015].

Arkin, R. C., 1998. *Behavior-based Robotics*. 1st Edition ed. Cambridge(Massachusetts): MIT Press.

Aspelund, K., 2015. *The Design Process*. 3rd Edition ed. New York(NY): Bloomsbury Publishing Inc.

Association for Robots in Architecture, 2013. *Robotic Fabrication in Architecture, Art and Design.* Wien/New York: Springer.

Association for Robots in Architecture, 2015. *KUKA|prc* [Online]
Available at: http://www.robotsinarchitecture.org/kuka-prc
[Accessed 5 November 2015].

Ayres, R. & Miller, S., 1981. *The Impacts of Industrial Robots*, Pittsburgh, PA: s.n.

Bain, A., 1855. *The senses and the intellect*. Washington(DC): University Publications of America.

Bauer, A., Wollherr, D. & Buss, M., 2007. Human-Robot Collaboration: A Survey. *International Journal of Humanoid Robotics.*

Bechthold, M. & King, N., 2013. Design Robotics. In: *Robotic Fabrication in Architecture, Art, and Design*. Wien/New York: Springer, pp. 118-129.

Blattner, M. M., Sumikawa, D. A. & Greenberg, R. M., 1989. Earcons and Icons: Their structure and common design principles. *Human-Computer Interaction*, 4(1), pp. 11-44.

bocoup, 2015. *Johnny-Five: The JavaScript Robotics Programming Framework.* [Online]
Available at: http://johnny-five.io/
[Accessed 17 October 2015].

Breazeal, C. et al., 2005. *Effects of Nonverbal Communication on Efficiency and Robustness in Human-Robot Teamwork*. s.l., IEEE, pp. 708-713.

Brell-Cokcan, S. & Braumann, J., 2010. *A New Parametric Design Tool for Robot Milling*. New York, NY, ACADIA, pp. 357-363.

Brell-Cokcan, S. & Braumann, J., 2011. *Parametric Robot Control: Integrated CAD/ CAM for Architectural Design*. Banff, AB, ACADIA, pp. 242-251.

Brennan, S. & Hulteen, E., 1995. *Interaction and Feedback in a Spoken Language System: A Theoretical Framework*. s.l.:Knowledge-Based Systems.

Bruemmer, D. J., Dudenhoeffer, D. D. & Marble, J. L., 2002. Dynamic-Autonomy for Urban Search and Rescue. *AAAI Mobile Robot Workshop*, pp. 1-18.

Bryant, S. L., Forte, A. & Bruckman, A., 2005. *Becoming Wikipedian: Transformation of Participation in a Collaborative Online Encyclopedia*. New York, ACM, pp. 1-10.

Buchenau, M. & Suri, J. F., 2000. *Experience Prototyping*. New York, ACM, pp. 424-433.

Chang, W.-L., Sabanovic, S. & Huber, L., 2013. *Use of Seal-like Robot {PARO} in Sensory Group Therapy for Older Adults with Dementia*. Piscataway, IEEE Press, pp. 101-102.

Clark, H. H., 1996. *Using Language*. s.l.:Cambridge University Press.

Cohen, P. & Levesque, H., 1991. Teamwork. *Cognitive Science and Artificial Intelligence*, September, 25(4), pp. 487-512.

Cote, N., Canu, A., Bouzid, M. & Mouaddib, A.-I., 2012. *Humans-Robots Sliding Collaboration Control in Complex Environments with Adjustable Autonomy.* Washington DC, IEEE Computer Society, pp. 146-153.

Csikszentmihalyi, M., 2013. *Creativity: the Psychology of Discovery and Invention*. Modern Classics Edition ed. New York(NY): Harper Perennial.

Daas, M., 2014. Toward a Taxonomy of Architectural Robotics. *Proceedings of the XVIII Conference of the Iberoamerican Society of Digital Graphics: Design in Freedom*, 1(8), pp. 623-626.

Dautenhahn, K. et al., 2009. KASPAR-a minimally expressive humanoid robot for human-robot interaction research. *Applied Bionics and Biomechanics*, 6(34), pp. 369-397.

Dörfler, K., Rist, F. & Rust, R., 2013. Interlacing. In: J. R. O'Donovan, ed. *Robotic Fabrication in Architecture, Art, and Design*. Wien/New York: Springer, pp. 82-91.

Decoded, 2015. DecodedCo/leap-arduino. [Online]
Available at: https://github.com/DecodedCo/leap-arduino
[Accessed 16 October 2015].

DiSalvo, C. et al., 2010. CarlDiSalvo. [Online]
Available at: http://carldisalvo.com/posts/growbot-garden/

[Accessed 14 December 2015].

Droplet, 2014. Home. [Online]
Available at: http://smartdroplet.com/
[Accessed 4 December 2015].

Dyson, 2015. Dyson 360 Eye™ robot. [Online]
Available at: https://www.dyson360eye.com/
[Accessed 4 December 2015].

Fago, M., 2010. cyberpro4/openshowvar. [Online]
Available at: https://github.com/cyberpro4/openshowvar
[Accessed 28 November 2015].

Firmata, 2015. *arduino/StandardFirmata.ino at master • firmata/arduino*. [Online]
Available at: https://github.com/firmata/arduino/blob/master/examples/StandardFirmata/StandardFirmata.ino
[Accessed 20 December 2015].

Firmata, 2015. *firmata/protocol.* [Online]
Available at: https://github.com/firmata/protocol
[Accessed 20 December 2015].

Gamwell, L., 2015. *Mathematics and Art: A Cultural History*. 1st Edition ed. Woodstock(Oxfordshire): Princeton University Press.

Goodrich, M. A. & Schultz, A. C., 2007. Human-robot Interaction: A Survey. *Foundations and Trends in Human-Computer Interaction*, February, 1(3), pp. 203-275.

Goodrich, M. A. & Schultz, A. C., 2007. Human-Robot Interaction: A Survey. *Foundations and Trends in Human-Computer Interaction*, 1(3), pp. 203-275.

Goza, S. M., Ambrose, R. O., Diftler, M. A. & Spain, I. M., 2004. *Telepresence control of the NASA/DARPA robonaut on a mobility platform*. New York, ACM, pp. 623-629.

Gramazio Kohler Research, ETH Zurich, 2011. *Procedural Landscapes 1*. [Online]
Available at: http://gramaziokohler.arch.ethz.ch/web/e/lehre/208.html
[Accessed 25 December 2015].

Gramazio Kohler Research, 2006. *Winery Gantenbein*.[Online]
Available at: http://gramaziokohler.arch.ethz.ch/web/e/forschung/52.html
[Accessed 25 December 2015].

Gramazio Kohler Research, 2011. *The Endless Wall.* [Online]
Available at: http://gramaziokohler.arch.ethz.ch/web/e/forschung/216.html
[Accessed 25 December 2015].

Gramazio Kohler Research, 2012. *Stratifications.* [Online]
Available at: http://gramaziokohler.arch.ethz.ch/web/e/forschung/206.html
[Accessed 25 December 2015].

Gramazio, F. & Kohler, M., 2014. *Made by Robots: Challenging Architecture at a Larger Scale*. 1st Edition ed. s.l.:AD.

Gramazio, F. & Kohler, M., 2015. Digital Materiality in Architecture. In: *Digital Materiality in Architecture*. s.l.:s.n.

Gramazio, F., Kohler, M. & Willmann, J., 2014. *The Robotic Touch: How Robots Change Architecture*. 1st Edition ed. s.l.:Park Books.

Grasshopper, 2015. *Grasshopper3d*. [Online]
Available at: http://www.grasshopper3d.com/
[Accessed 21 December 2015].

Green, S., Billinghurst, M., Chen, X. & Chase, G., 2007. Human-Robot Collaboration: A Literature Review and Augmented Reality Approach. *Advanced Robotic Systems*, pp. 1-18.

Grosz, B., 1996. Collaborative Systems.

Haidegger, T., Sandor, J. & Benyo, Z., 2011. Surgery in space: the future of robotic telesurgery. *Surgical Endoscopy*, March, 25(3), pp. 681-690.

HAL Robotics, 2015. *HAL-Robotics.* [Online]
Available at: http://www.hal-robotics.com/
[Accessed 21 December 2015].

Hoffman, G. & Breazeal, C., 2004. *Collaboration in Human-Robot Teams*. s.l., American Institute of Aeronautics and Astronautics, pp. 1-18.

Hoven, E. V. D. & Mazalek, A., 2010. Grasping Gestures: Gesturing with Physical Artifacts. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 25(S3), pp. 255-271.

Husqvarna, 2015. *Husqvarna robot lawn mower, automatic lawn mower, automower.* [Online]
Available at: http://www.husqvarna.com/us/products/robotic-lawn-mowers/
[Accessed 4 December 2015].

Leap Motion Inc., 2012. *Leap Motion*. [Online]
Available at: https://www.leapmotion.com/
[Accessed 1 December 2015].

Interbots, 2012. *Popchilla's World: A Game for Kids with Autism.* [Online]
Available at: http://www.popchillasworld.com/
[Accessed 4 December 2015].

International Federation of Robotics, 2015. *Products.* [Online]
Available at: http://www.ifr.org/industrial-robots/products/
[Accessed 4 December 2015].

International Federation of Robotics, 2015. *Statistics.* [Online]
Available at: http://www.ifr.org/industrial-robots/statistics/
[Accessed 3 12 2015].

iRobot, 2015. *iRobot Roomba Vacuum Cleaning Robot.* [Online]
Available at: http://www.irobot.com/For-the-Home/Vacuum-Cleaning/Roomba.aspx
[Accessed 04 December 2015].

Ishii, H. & Ullmer, B., 1997. *Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms*. New York, ACM, pp. 234-241.

Jain, J., Lund, A. & Wixon, D., 2011. *The Future of Natural User Interfaces.* New York, ACM.

Johns, R. L., 2013. Augmented Reality and the Fabrication of Gestural Form. In: J. R. O'Donovan, ed. *Robotic Fabrication in Architecture, Art, and Design*. Wien/New York: Springer, pp. 248-255.

Kaber, D. B. & Endsley, M. R., 2004. The Effects of Level of Automation and Adaptive Automation on Human Performance, Situation Awareness and Workload in a Dynamic Control Task. *Theoretical Issues in Ergonomics Science*, 5(3), pp. 113-153.

Kaltenbrunner, M., 2015. *TUIO Protocol Specification.* [Online]
Available at: http://www.tuio.org/?specification
[Accessed 17 December 2015].

Kaltenbrunner, M., 2015. *TuioListener.* [Online]
Available at: http://www.tuio.org/api/java/TUIO/TuioListener.html
[Accessed 18 December 2015].

Kaltenbrunner, M. & Bencina, R., 2007. *reacTIVision: A Computer-vision Framework for Table-based Tangible Interaction*. Baton Rouge, Louisiana, ACM, pp. 69-74.

Kidd, C. D., Taggart, W. & Turkle, S., 2006. *A sociable robot to encourage social interaction among the elderly.* Orlando, FL, IEEE, pp. 3972-3976.

Kim, M. J. & Maher, M. L., 2008. *The Impact of Tangible User Interfaces on Spatial Cognition during Collaborative Design*. Design Studies, 29(3), pp. 222-253.

Kirsh, D. & Maglio, P., 1994. *On Distinguishing Epistemic from Pragmatic Action. Cognitive Science*, 18(4), pp. 513-549.

KUKA Roboter GmbH, 2015. *Data Sheet | KR 6 R700 sixx*. Augsburg: KUKA Roboter GmbH.

KUKA Robotics, 2006. *KUKA Industrial Robots - KUKA Robotics introduces new family of SCARA robots.* [Online]
Available at: http://www.kuka-robotics.com/usa/en/pressevents/productnews/NN_060713_New_Family_Of_SCARA_Robots.htm
[Accessed 3 December 2015].

KUKA Robotics, 2010. *KUKA System Software 5.5: Operating and Programming Instructions for System Integrators*. Augsburg: KUKA Roboter GmbH.

KUKA Robotics, 2014. *KR C4 compact - Operating Instructions.* Augsburg: KUKA Roboter GmbH.

Kuka Robotics, 2014. *KUKA Industrial Robots - LBR iiwa*. [Online]
Available at: http://www.kuka-robotics.com/en/products/industrial_robots/sensitiv/
[Accessed 18 October 2015].

KUKA Robotics, 2015. *KUKA Industrial Robots - KR 180 R2500 extra (KR QUANTEC extra).* [Online]
Available at: http://www.kuka-robotics.com/usa/en/products/industrial_robots/high/extra/kr180_r2500_extra/start.htm
[Accessed 3 December 2015].

KUKA Robotics, 2015. *KUKA Industrial Robots - KR 6 R700 sixx (KR AGILUS).* [Online]
Available at: http://www.kuka-robotics.com/en/products/industrial_robots/small_robots/kr6_r700_sixx/
[Accessed 14 November 2015].

KUKA Robotics, 2015. *KUKA Industrial Robots - KR C4 compact.* [Online]
Available at: http://www.kuka-robotics.com/en/products/controllers/kr_c4_compact/
[Accessed 2 December 2015].

KUKA Robotics, 2015. *KUKA Industrial Robots - KUKA smartPAD*. [Online]
Available at: http://www.kuka-robotics.com/en/products/controllers/smartPAD/
[Accessed 2 December 2015].

KUKA Robotics, 2015. *Specification | KR 6 R700 sixx.* [Online]
Available at: http://www.kuka-robotics.com/res/sps/e6c77545-9030-49b1-93f5-4d17c92173aa_Spez_KR_AGILUS_sixx_en.pdf
[Accessed 25 October 2015].

KUKA, 2015. *KUKA.WorkVisual.* [Online]
Available at: http://www.kuka-robotics.com/en/products/software/engineering_environment/
[Accessed 3 December 2015].

Kvale, S., 2007. *Doing Interviews*. 1st Edition ed. London: SAGE Publications Ltd.
Kwon, W. Y. & Suh, I. H., 2011. Towards Proactive Assistant Robots for Human Assembly Tasks. New York, ACM, pp. 175-176.

Laursen, C. Ø., Pedersen, S., Merritt, T. & Caprani, O., 2015. Robot-Supported Food Experiences: Exploring Aesthetic Plating with Prototypes. In: *Cultural Robotics*. s.l.:IEEE Springer.

Leap Motion, 2015. *Frames — Leap Motion JavaScript SDK v2.3 documentation.* [Online]
Available at: https://developer.leapmotion.com/documentation/javascript/devguide/Leap_Frames.html
[Accessed 20 December 2015].

Leap Motion, 2015. *Hand — Leap Motion JavaScript SDK v2.3 documentation.* [Online]
Available at: https://developer.leapmotion.com/documentation/javascript/api/Leap.Hand.html#Hand.pinchStrength
[Accessed 20 December 2015].

Lovchik, C. S. & Diftler, M. A., 1999. *The Robonaut Hand: A Dexterous Robot Hand For Space*. Detroit, IEEE, pp. 907-912.

Lupetti, M. L., Rosa, S. & Ermacora, G., 2015. *From a Robotic Vacuum Cleaner to Robot Companion: Acceptance and Engagement in Domestic Environments.* New York, ACM, pp. 119-120.

MacKinnon, D. W., 1962/1978. *In search of human effectiveness.* New York(NY): Universe Books.

McCarthy, J., 1979. *Ascribing Mental Qualities to Machines.* s.l.:Stanford University.

MeArm Robotics, 2015. *MeArm Robotics - Home of the Original MeArm Project.* [Online]
Available at: http://mearm.com/
[Accessed 20 December 2015].

Merriam-Webster, 2015. *Automaton.* [Online]
Available at: http://www.merriam-webster.com/dictionary/automaton
[Accessed 26 November 2015].

MICROPIK, 2015. *SG90Servo.pdf* [Online]
Available at: http://www.micropik.com/PDF/SG90Servo.pdf
[Accessed 30 September 2015].

Microsoft, 2015. *Kinect - Windows app development.* [Online]
Available at: https://dev.windows.com/en-us/kinect
[Accessed 3 January 2016].

Microsoft, 2015. *Windows Embedded 7 | Microsoft.* [Online]
Available at: https://www.microsoft.com/windowsembedded/en-us/windows-embedded-7.aspx
[Accessed 21 December 2015].

Minamizawa, K. et al., 2012. *TECHTILE Toolkit: A Prototyping Tool for Designing Haptic Media*. Los Angeles, CA, ACM, p. 22.

NASA, 2013. *MERLithograph.pdf*. [Online]
Available at: http://mars.nasa.gov/mer/home/resources/MERLithograph.pdf
[Accessed 3 December 2015].

NASA, 2013. *Robonaut Operates Task Board in Space | NASA.* [Online]
Available at: http://www.nasa.gov/multimedia/imagegallery/image_feature_2423.html
[Accessed 3 December 2015].

NASA, 2015. *Question About the Flight Plan*. [Online]
Available at: http://mars.jpl.nasa.gov/mgs/faqs/faq_mplan.html
[Accessed 01 December 2015].

Neilson, T., 2005. *Mars Exploration Rovers Surface Fault Protection*. Waikoloa, IEEE, pp. 14-19.

Nintendo Co., L., 2015. *Nintendo.* [Online]
Available at: http://wii.com/
[Accessed 1 December 2015].

Node.js Foundation, 2015. [Online]
Available at: https://nodejs.org/en/
[Accessed 17 September 2015].

Norman, D., 2010. Natural User Interfaces Are Not Natural. *Interactions*, 17(3), pp. 6-10.

Novikova, J. & Watts, L., 2014. *A Design Model of Emotional Body Expressions in Non-humanoid Robots*. s.l., ACM.

Odico Aps, 2015. *Odico.* [Online]
Available at: http://www.odico.dk/
[Accessed 2 January 2016].

Oracle, 2015. *KeyListener (Java Platform SE 7).* [Online]
Available at: http://docs.oracle.com/javase/7/docs/api/java/awt/event/KeyListener.html
[Accessed 18 December 2015].

Oracle, 2015. *MouseMotionListener (Java Platform SE 7).* [Online]
Available at: http://docs.oracle.com/javase/7/docs/api/java/awt/event/MouseMotionListener.html
[Accessed 22 December 2015].

Paek, T. & Horvitz, E., 1999. *Uncertainty, Utility, and Misunderstanding: A Decision-Theoretic Perspective on Grounding in Conversational Systems.* s.l.:Psychological Models of Communication in Collaborative Systems.

Parasuraman, R., Sheridan, T. B. & Wickens, C. D., 2000. A Model for Types and Levels of Human Interaction with Automation. *IEEE Transactions on System, Man, and Cybernetics - Part A: Systems and Humans*, May, 30(3), pp. 286-297.

Pavlovic, V., Sharma, R. & Huang, T., 1997. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), pp. 677-695.

P. Chris Leger, A. T.-O. J. R. W. S. A. M. R. G. B. J. J. B. F. R. H. B. K. C. E. T. B. M. W. M., 2005. *Mars Exploration Rover surface operations: driving spirit at Gusev Crater.* Waikoloa, IEEE, pp. 1815-1822.

Phillips, E., Ososky, S., Grove, J. & Jentsch, F., 2011. From Tools to Teammates: Toward the Development of Appropriate Mental Models for Intelligent Robots. *Proceedings of the Human Factors and Ergonomics Society*, pp. 1491-1495.

Pineau, J. et al., 2003. Towards Robotic Assistants in Nursing Homes: Challenges and Results. *Robotics and Autonomous Systems*, March, 42(3-4), pp. 271-281.

Quek, F., 1995. Eyes in the Interface. *Image and Vision Computing*, August. Volume 13.

Reed, S. E. et al., 2014. Shaping Watersheds Exhibit: An Interactive, Augmented Reality Sandbox for Advancing Earth Science Education. *AGU Fall Meeting Abstracts*, December.

Riskin, J., 2003. The Defecating Duck, or, the Ambiguous Origins of Artificial Life. *Critical Inquiry*, 29(4), pp. 599-633.

Robert McNeel & Associates, 2015. *Rhino3D.* [Online]
Available at: https://www.rhino3d.com
[Accessed 21 December 2015].

Rosheim, M. E., 2006. *Leonardo's Lost Robots.* Berlin: Springer.

Sabanovic, S., Bennett, C. C., Wan-Ling, C. & Huber, L., 2013. *PARO robot affects diverse interaction modalities in group sensory therapy for older adults with dementia*. Seattle, WA, IEEE, pp. 1-6.

Sabanovic, S., Bennett, C. & Lee, H. R., 2014. *Towards Culturally Robust Robots: A Critical Social Perspective on Robotics and Culture*. s.l., ACM/IEEE.

Samani, H. et al., 2013. Cultural Robotics: The Culture of Robotics and Robotics in Culture. *International Journal of Advanced Robotic Systems*, 10(400).

Sanfilippo, F. et al., 2014. *JOpenShowVar: An open-source cross-platform communication interface to Kuka robots.* Hailar, Hulun Buir, IEEE, pp. 1154-1159.

Sawyer, R. K., 2012. *Explaining Creativity: The Science of Human Innovation*. 2nd Edition ed. Oxford: Oxford University Press.

Scholtz, J., 2002. *Theory and Evaluation of Human Robot Interactions*. Hawaii, IEEE.

Sharlin, E. et al., 2004. On Tangible User Interfaces, Humans and Spatiality. *Personal and Ubiquitous Computing*, September, 8(5), pp. 338-346.

Shibata, T. et al., 2001. *Mental commit robot and its application to therapy of children*. Como, IEEE, pp. 1053-1058.

Shi, J., Jimmerson, G., Pearson, T. & Menassa, R., 2012. *Levels of Human and Robot Collaboration for Automotive Manufacturing.* New York, ACM, pp. 95-100.

Shneiderman, B., 1983. *Direct Manipulation: A Step Beyond Programming* Languages. Computer, August, 16(8), pp. 57-69.

Shneiderman, B., 1997. *Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces*. New York, ACM, pp. 33-39.

SRI International, 1984. *Shakey the Robot,* Menlo Park, CA: SRI International.

Stiehl, W. D. et al., 2009. *The Chanty Bear: Toward a Robot Teaching English to Children.* New York, ACM, pp. 317-320.

Stubbs, K., Wettergreen, D. & Hinds, P. J., 2007. Autonomy and Common Ground in Human-Robot Interaction: A Field Study. *Intelligent Systems*, March, 22(2), pp. 42-50.

Tesla, N., 1898. *Method of and apparatus for controlling mechanism of moving vessels or vehicles*. United States of America, Patent No. US613809 A.

Tesla, N., 1900. The Problem of Increasing Human Energy. *The Century Illustrated Monthly Magazine*, June, LX(2), p. 320.

Tetris, 1984. *Tetris.* [Online]
Available at: http://tetris.com/
[Accessed 11 December 2015].

Toshiba Machine Company, 2015. *Toshiba Machine Co., America - System Robotics.* [Online]
Available at: http://www.toshiba-machine.com/csproductcatalog.aspx?dept=4
[Accessed 3 December 2015].

Universal Robots, 2015. *UR Download | Support Site.* [Online]
Available at: http://www.universal-robots.com/download/?option=16594
[Accessed 3 December 2015].

Wada, K. & Shibata, T., 2007. Living With Seal Robots - Its Sociopsychological and Physiological Influences on the Elderly at a Care House. *IEEE Transactions on Robotics*, 8 October, 23(5), pp. 972-980.

Wada, K. et al., 2005. *Psychological and Social Effects of One Year Robot Assisted Activity on Elderly People at a Health Service Facility for the Aged*. Barcelona, Spain, IEEE, pp. 2785-2790.

Wada, K., Shibata, T., Saito, T. & Tanie, K., 2004. Effects of robot-assisted activity for elderly people and nurses at a day service center. *Proceedings of the IEEE*, 25 October, 92(11), pp. 1780-1788.

Wigdor, D. & Wixon, D., 2011. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. 1st Edition ed. San Francisco: Morgan Kaufmann Publishers Inc..

Wikipedia, 2015. *Adobe Illustrator - Wikipedia*. [Online]
Available at: https://en.wikipedia.org/wiki/Adobe_Illustrator
[Accessed 27 December 2015].

Wikipedia, 2015. *Adobe Photoshop - Wikipedia*. [Online]
Available at: https://en.wikipedia.org/wiki/Adobe_Photoshop
[Accessed 27 December 2015].

Wikipedia, 2015. *da Vinci Surgical System - Wikipedia.* [Online]
Available at: https://en.wikipedia.org/wiki/Da_Vinci_Surgical_System
[Accessed 4 Januar 2016].

Wikipedia, 2015. *Digesting Duck - Wikipedia.* [Online]
Available at: https://en.wikipedia.org/wiki/Digesting_Duck
[Accessed 3 December 2015].

Wikipedia, 2015. *Leap Motion - Wikipedia.* [Online]
Available at: https://en.wikipedia.org/wiki/Leap_Motion#Technology
[Accessed 20 December 2015].

Wiktionary, 2015. *Robota*. [Online]
Available at: https://en.wiktionary.org/wiki/robota
[Accessed 3 December 2015].

Willis, K. D. et al., 2011. *Interactive Fabrication: New Interfaces for Digital Fabrication*. New York, ACM, pp. 69-72.

Willmann, J., Gramazio, F., Kohler, M. & Langenberg, S., 2013. Digital by Material. In: J. R. O'Donovan, ed. *Robotic Fabrication in Architecture, Art, and Design*. Wien/New York: Springer, pp. 12-27.

Wind River, 2015. *VxWorks.* [Online]
Available at: http://windriver.com/products/vxworks/
[Accessed 21 December 2015].

YouTube, 2011. *RobotStudio Tutorials - How to make a writing program in RobotStudio - part 2 - YouTube.* [Online]
Available at: https://www.youtube.com/watch?v=NxI3h6u8wk0
[Accessed 8 January 2016].

YouTube, 2011. *Tutorial Kuka Sim Pro Gatarito77 - YouTube.* [Online]
Available at: https://www.youtube.com/watch?v=2cRAwmpSKkE
[Accessed 8 January 2016].

Zimmerman, J., Forlizzi, J. & Evenson, S., 2007. *Research Through Design As a Method for Interaction Design Research in HCI*. San Jose, CA, ACM, pp. 493-502.

Appendix

List of Full-page Images used in the Book:

[Chapter 1 Ending]
http://scm.dk/sites/default/files/UpdateB07_2.jpg

[Chapter 2 Introduction]
http://michaelmurray.ca/wp-content/uploads/2015/03/robot-toy.jpg

[Chapter 2 Ending]
http://www.robarch2016.org/wp-content/uploads/2015/02/Fig_6a.jpg

[Chapter 4 Introduction]
http://www.hotelmanagement.com.au/wp-content/uploads/2012/10/Ibis-Robot-1.jpg

[Chapter 4 Ending]
https://www.espazium.ch/schalungen-digital-formen

[Chapter 5 Introduction]
http://dfabclass.com/fall_15/wp-content/uploads/2015/10/IF_teaser_web.jpg

[Chapter 8 Introduction]
http://f.fastcompany.net/multisite_files/fastcompany/poster/2015/06/3047350-poster-p-1-this-robot-can-3-d-print-a-steel-bridge-in-mid-air.jpg

[Bibliography Introduction]
http://whytoread.com/wp-content/uploads/2015/03/best-books-book-youll-ever-read.jpg

APPENDIX 1 - INTERVIEW GUIDE FOR MASTER AND PH.D.

Interview guide for interview with a Master's student and Ph.D. student from Aarhus School of Architecture. Corresponding transcription can be found on the DVD medium as "Transcription - Appendix 1".

Semi-structured interview, these are only guiding questions and may be different from the ones asked during the interview. In addition, since it was a semi-structured interview, follow up questions may appear in the recorded version.

Can you describe how your process is working with the robot?
  Is it a iterative process?

How do you incorporate the robot in your overall design process?
  What advantages does it have?

How was your process before the robot were available to you?

How would you describe the interaction with the robot?
  If you had to describe the role of the robot - what roles does it take then?

How do you envision the use of the robot in the future of architecture?

How would you prefer interacting with the robot?
  Could you imagine a better/more efficient way to interact with the robot?
  More free-hand-like approach?

What are your view on robotic technology being a more integrated part of our lives?

APPENDIX 2 - INTERVIEW GUIDE FOR JOHANNES BRAUMANN

Interview guide for interview with Johannes Braumann from Association for Robots in Architecture. Corresponding transcription can be found on the DVD medium as "Transcription - Appendix 2".

Semi-structured interview, these are only guiding questions and may be different from the ones asked during the interview. In addition, since it was a semi-structured interview, follow up questions may appear in the recorded version.

- What is the role of the robot in the field of architecture?

- Where/How does the robot influence or complement the architect's design process?

- How do you envision the use of the robot in the future?

- What are the potentials for robots, working collaboratively with humans, in the field of architecture?

- How do you see the creativity unfold in the interaction with human, robot and material?

- What are your future vision for how architects interact with robots?
More focus on the physical world instead of the digital world?

APPENDIX 3 - INTERVIEW GUIDE FOR MICHAEL KNAUSS

Interview guide for interview with Michael Knauss from ETH Zürich. Corresponding transcription can be found on the DVD media as "Transcription - Appendix 3".

Semi-structured interview, these are only guiding questions and may be different from the ones asked during the interview. In addition, since it was a semi-structured interview, follow up questions may appear in the recorded version.

- How do you incorporate robots in your design process - can you walk us through a "typical" design process, step by step? E.g. Pike Loop on Manhattan?

- How/Where does the robot influence or complement the architect's design process?

- What are the potentials for robots, working collaboratively with humans, in the field of Architecture?

- How do you see the robot acting as an independent collaborator?

- What limitations do you see/or have experienced working with robots?

- How do you see creativity unfold in the interaction with human, robot and material?

- Can you envision how the robot could alter the programmed design or otherwise contribute to the ideas of the Architect?

- How does the craftmanship translate into a product through the actions of a robot?

- How do you envision Architects, or creative people in general, interact with robots?