NTNU – Trondheim
Norwegian University of
Science and Technology

Advanced Remote Control of Industrial Robots

Fredrik Reme

Master of Science in Product Design and Manufacturing
Submission date: June 2012
Supervisor: Trygve Thomessen, IPK

Norwegian University of Science and Technology
Department of Production and Quality Engineering

NTNU
Faculty of Engineering Science and Technology
Department of Production and Quality Engineering

Date	Our reference
2012-02-03	TTH/LMS

MASTER THESIS
2012
for
stud. techn. Fredrik Reme

ADVANCE REMOTE CONTROL OF INDUSTRIAL ROBOTS
(Fjernstyring av industrirobotsystemer ved hjelp av multimodal bruker-kommunikasjon)

TASK TO BE SOLVED

Traditionally, industrial robot systems have been installed in big companies due to the fact that the level of operation requires knowledge, training and close support from experts to be successful. However, during the last years, the trend has changed, and industrial robots are becoming more common in general industry, thus; small and medium sized companies (SME). This trend is putting new challenges on how to provide immediate and cost efficient knowledge transfer and support from the system integrators to the end customers.

This master thesis is going to focus on the development of new methods and technology for doing the major part of the end customer support remotely. The solutions are intended to focus upon how to efficiently transfer and communicate the state of the physical robot cell to a remote user. The theoretical foundation behind this is Cogntive Informatics Communication, CogInfoCom. The goal is to setup a full remote operation system, and remote control the robot system from another physical location.

The solutions shall be developed in PPM's laboratory in Trondheim, for NACHI MR20 industrial robot, with PPM's high speed USB interface to NACHI's AX20 controller.

The implementation shall be done using a standard personal computer (PC) communicating with the OLIMEX and PC attached to the MR20 robot. In addition, there will be a Kinect motion sensing input device. The following tasks are to be accomplished:

i. Introductory study of the need for and the challenges to remote operate industrial robot systems. Discuss how remote control can be simplified/assisted by real time sensor feedback on the physical robot system.
ii. Introductory study of teach-in programming using wireless devices. Develop methodology and system structure for this.

iii. Introductory literature study of CogInfoCom and considerations about the appliance of it in advance remote control of industrial robots. Discuss how CogInfoCom can be used efficiently to communicate between the remote operator and the physical robot system.
iv. Develop and specification and a structure for the information channel between the remote operation system and the physical robot system.
v. Develop the system structure of the remote operation system based upon CogInfoCom. This includes the teach-in programming using Kinect.
vi. Develop the laboratory setup of the remote operation system based upon CogInfoCom. This includes the teach-in programming using Kinect.
vii. Experimental testing of the remote operation system.
viii. Documentation of the experimental setup including hardware and software.
ix. Documentation of the user functions to operate the experimental setup.
x. Documentation of the experimental results.

Within three weeks after the date of the task handout, a pre-study report shall be prepared. The report shall cover the following:

- An analysis of the work task's content with specific emphasis of the areas where new knowledge has to be gained.
- A description of the work packages that shall be performed. This description shall lead to a clear definition of the scope and extent of the total task to be performed.
- A time schedule for the project. The plan shall comprise a Gantt diagram with specification of the individual work packages, their scheduled start and end dates and a specification of project milestones.

The pre-study report is a part of the total task reporting. It shall be included in the final report. Progress reports made during the project period shall also be included in the final report.

The report should be edited as a research report with a summary, table of contents, conclusion, list of reference, list of literature etc. The text should be clear and concise, and include the necessary references to figures, tables, and diagrams. It is also important that exact references are given to any external source used in the text.

Equipment and software developed during the project is a part of the fulfilment of the task. Unless outside parties have exclusive property rights or the equipment is physically non-moveable, it should be handed in along with the final report. Suitable documentation for the correct use of such material is also required as part of the final report.

The candidate shall follow the work regulations at the company's plant. The candidate may not intervene in the production process in any way. All orders for specific intervention of this kind should be channelled through company's plant management.

The student must cover travel expenses, telecommunication, and copying unless otherwise agreed.

If the candidate encounters unforeseen difficulties in the work, and if these difficulties warrant a reformation of the task, these problems should immediately be addressed to the Department.

The assignment text shall be enclosed and be placed immediately after the title page.

Deadline: June 11th 2012.

Two bound copies of the final report and one electronic (pdf-format) version are required.

RESTRICTIONS
PPM has the rights to use the results from the Master Thesis for any purpose.

Responsible supervisor at NTNU and PPM AS:

Professor Trygve Thomessen
Managing Director
PPM AS
Leirfossveien 27
NO-7038 Trondheim
Norway
Phone: + 47 92 24 21 89
E-mail: tth@ppm.no

**DEPARTMENT OF PRODUCTION
AND QUALITY ENGINEERING**

Per Schjølberg

Per Schjølberg
Associate Professor/Head of Department

Trygve Thomessen
Responsible Supervisor

Summary

The last few years, the use of industrial robots for tasks such as material handling, welding, painting and assembly, has expanded considerably - also in general industry. A company's use of robots requires a certain expertise within the field of robotics. To hire in-house experts can be quite costly. Many small and medium sized enterprises (SMEs) consider robots in their productions, but cannot afford employing specialists. The challenge is therefore to find a way for robot system integrators to perform support and knowledge-transfer to the SMEs from the integrator's location, making it cheaper and more efficient. To address this, one possibility is to enable remote operation of the industrial robots. This thesis aims to develop such a system, where new methods and technologies are used to achieve end-customer support based on remote control. In developing a system like this, a major concern is how to efficiently communicate the state of the physical robot cell to the remote operator. The multidisciplinary field CogInfoCom (Cognitive Infocommunications) deals with the connections between cognitive sciences and info communications, and was used for solving problems regarding transmission of the robot's condition to the operator.

The system developed in this thesis centers around the use of a gripper analog that the operator moves in space to remotely control the robot. Auditory and visual feedback for communicating the state of the robot during control is used. The system includes functions for scaling, self-motion, motion guidance and teach-in programming, and was developed for a NACHI MR20 7-axis industrial robot. The combination of a MARG-sensor and a Microsoft Kinect was used to determine the gripper analog's position and orientation in space.

Practical experiments were performed for testing and verifying the scaling, the self-motion, the motion guidance and the teach-in programming. They demonstrated that all the functions lived up to their expectations. In addition, a test with a time-delay of one second was carried out. This resulted in indications that the control method aided the operator in time-delayed control, because it allowed him to plan motions ahead by relating them to his environment.

A qualitative test with five participants was done to test the system's general usability. All the participants performed a pick-and-place task, and everyone successfully completed this task. The observations during the usability test, suggested that the user interface was intuitive and easy to understand. However, the differences in camera-view perspective and control perspective could often cause confusions. With that said, it was clear that efficient usage of the system can be achieved with minimal training. During the usability test, it was also explored how the participants responded to a visual overlay, covering the operator's computer monitor, turning red. In this case, the overlay gradually turned red to warn the operator that a singularity was approaching. As most participants immediately halted the operation when this happened, the overlay concept was considered a beneficial way of conveying this information to the remote operator.

In conclusion, the system's utilization of a gripper analog as a tangible interface provides an intuitive way of remotely controlling industrial robots. Moreover, the use of CogInfoCom and multimodal man-machine communication, is considered an efficient way to communicate information of the robot's state to the operator.

Sammendrag

De siste årene har bruken av industrielle roboter økt betraktelig, også i generell industri. Roboter brukes ofte til oppgaver som materialhåndtering, sveising, spraymaling og montasje. For at en bedrift skal kunne benytte roboter, kreves det en viss kompetanse innen robotikk. Det å ansette eksperter innen feltet kan være kostbart. Mange små og mellomstore bedrifter (SMEs) ønsker å bruke roboter i sine produksjoner, men har ofte ikke råd til ansette spesialister. Det vil derfor være en utfordring for systemintegratorer å finne en måte å kunne utføre støtte og kunnskapsoverføring til SMEs fra integratorens kontor, noe som vil gjøre det billigere og mer effektivt. For å oppnå dette, kan man for eksempel muliggjøre fjernstyring av industriroboter. Denne masteroppgaven har som mål å utvikle et slikt fjernstyringssystem ved å ta i bruk nye metoder og teknologier for å oppnå slutt-kundestøtte basert på fjernstyring. Et viktig problem under utviklingen av et slikt system vil være hvordan man mest mulig effektivt kan formidle tilstanden til den fysiske robotcellen til en ekstern operatør. Det tverrfaglige feltet CogInfoCom (Cognitive Infocommunications) omhandler sammenhenger mellom kognitiv vitenskap og infokommunikasjon, og dette feltet ble benyttet for å løse problemer vedrørende formidling av robotens tilstand til operatøren.

Systemet som ble utviklet i denne masteroppgaven går ut på at operatøren fjernstyrer roboten ved å bevege en griperanalog i rommet. Det benyttes hørsels- og synsfeedback for å formidle robotens tilstand under styringen. Fjernstyringssystemet inkluderer funksjoner som skalering, selv-bevegelse, bevegelsesveiledning og robotprogrammering, og ble utviklet til en NACHI MR20 7-aksers industrirobot. Griperanalogens posisjon og orientering i rommet ble bestemt ved å kombinere en MARG-sensor og en Microsoft Kinect.

Det ble utført praktiske eksperimenter for å teste og verifisere skalering, selv-bevegelse, bevegelsesveiledning og robotprogrammering. Alle disse funksjonene levde opp til sine forventninger. I tillegg ble det gjort en test hvor det ble innført en tidsforsinkelse på ett sekund. Dette ga indikasjoner om at styringsmetoden hjalp operatøren i tidsforsinket styring, fordi det tillot ham å planlegge bevegelser ved å relatere dem til sine egne omgivelser.

Videre ble det gjennomført en kvalitativ test med fem deltakere for å undersøke systemets generelle brukbarhet. Deltakerne utførte en oppgave hvor de skulle ta opp og flytte en gjenstand ved fjernstyring av industriroboten. Alle klarte å gjennomføre denne oppgaven. Observasjonene fra denne testen viste at brukergrensesnittet var intuitivt og lett å forstå. Imidlertid skapte forskjellene i kameraperspektivet og styringsperspektivet ofte forvirring. Det var likevel tydelig at det kun trengs minimal opplæring for å klare å bruke systemet på en

effektiv måte. Under brukbarhetstesten ble det også undersøkt hvordan forsøkskandidatene reagerte når et visuelt overlag på styringsdataskjermene ble rødt. I dette tilfellet ble overlaget gradvis rødt for å varsle operatøren om at en singularitet nærmet seg. Ettersom de fleste deltakerne umiddelbart stanset robotstyringen når dette skjedde, ble konseptet ansett som en gunstig måte å formidle denne informasjonen til operatøren.

Det konkluderes med at systemets bruk av en griperanalog er en intuitiv måte å fjernstyre industriroboter på. Videre er bruken av CogInfoCom og multimodal bruker-kommunikasjon ansett som effektivt ved formidling av informasjon om robotens tilstand til operatøren.

Preface

This is a master's thesis in production systems, which is a part of an integrated master's degree in mechanical engineering at NTNU, Trondheim.

The thesis was carried out by stud.techn. Fredrik Reme on behalf of PPM AS and NTNU.

The problem description has been developed by Professor Trygve Thomessen, and concerns advanced remote control of industrial robots. After discussion with Prof. Thomessen, the problem description has been slightly modified compared to the original version. Due to the amount of work related to the practical implementations, parts of the theoretical studies pertaining to *CogInfoCom* and *the needs and challenges in remote operation* has been given lower priority, and should be evaluated as such. In addition, testing the system from another physical location is no longer a criterion.

The thesis will be evaluated on the basis of a written report, as well as other material pertaining to it.

Trondheim, June 4, 2012
Fredrik Reme
Fredrik Reme

Approval for the changes in the problem description:

Trondheim, June 4, 2012

Prof. Trygve Thomessen

Acknowledgment

I would like to take the opportunity to thank Professor Trygve Thomessen for his great help, support and encouragement through the semester. I would also like to thank him for giving me access to PPM AS' robot laboratories, and for placing important equipment at my disposal.

I would also like to thank stud.techn Audun R. Sanderud, PhDc Balazs Daniel, PhDc Tomasz Kosicki and MSc Laszlo Nagy for both technical help and help in discussion.

In addition, I would like to thank my brother, Bjørn-Atle Reme, for valuable input and proofreading. Lastly, I would like to thank Trine Skuland for her moral support, patience and help during the work with my thesis.

Fredrik Reme

Contents

Chapter 1

Introduction

1.1 Background

The increasing labour costs in the western countries the last years has changed the landscape for traditional production companies. It has led to a push for increased use of industrial robots in a growing set of industrial tasks. While large enterprises can afford to have in-house staff with competence and expertise within the field of robotics and production automation, this may not be the case for small and medium sized enterprises (SMEs) (SMErobot, 2012).

The discrepancy between the SMEs' need for advanced technology like robotics while having limited competence, provides new challenges for the system integrators and their end customers. Much of the challenge revolves around handling the SMEs higher requirements for support and knowledge-transfer to enable them to sufficiently utilize an industrial robot system.

One possible approach for tackling this problem, is for the industrial robotic systems to be able to be remotely operated. This is based on the premise that if one can perform remote assistance for the SMEs, the enterprise can get faster and cheaper support irrespective of the geographical distance between the enterprise and the system integrator or other supportive services.

There are a lot of challenges related to efficient remote operation of industrial robots. Humans are highly dependent on using all of their trained senses while performing every-day tasks. In robot operation for example, stereo- and peripheral vision is important for depth perception and spatial awareness, which make the operator able to skillfully control the robot around obstacles and singularities etc. During remote control, however, some of the perceptional information is lost, causing effective control to become more challenging.

In tackling the challenge of efficiently communicating the state of the physical robot cell to a remote user, the field of Cognitive Infocommunications (CogInfoCom), can be applied. The primary goal of Cognitive Infocommunications according to Baranyi and Csapo (2010) is:

> *Provide a complete view of how brain processes can be merged with info communications devices so that the cognitive capabilities of the human brain may not only be efficiently extended through these devices, irrespective of geographical distance, but may also be efficiently matched with the capabilities of any artificially cognitive system.*

This thesis aims to explore if the use of the theoretical foundations of CogInfoCom combined with natural and intuitive control methods, can aid in facing the challenges of providing instant support and knowledge-transfer to the SMEs via remote operation.

1.2 Problem Formulation

The focus of this master thesis is on development of new methods and technology for enabling robot integrators to perform end-customer support remotely. A remote operation system based upon CogInfoCom is to be developed for a NACHI MR20 robot at PPM's laboratory in Trondheim. The remote operation system should include teach-in programming through the use of the Microsoft Kinect sensor.

Objectives
To meet the goals of the problem description, the following tasks has to be accomplished:

1. Introductory study of the need for and challenges to remote operate industrial robot systems.
2. Introductory study of teach-in programming using wireless devices.
3. Introductory study of CogInfoCom and considerations about the appliance of it in advanced remote control of industrial robots.
4. Develop a specification and structure for the information channel between the remote operation system and physical robot system.
5. Develop a system structure for the remote operation system.
6. Develop the laboratory setup of the remote operation system.
7. Experimental testing of the remote operation system.
8. Documentation of the experimental setup, including hardware and software.
9. Documentation of the user functions to operate the experimental setup.
10. Documentation of the experimental results.

1.3 Available Hardware

This section presents the available hardware during for the thesis. Datasheets can be found in appendix H.

Industrial Robot Arm

The industrial robot available for use during the work was a NACHI MR20 7-axis industrial robot with a NACHI AX20 Controller.

Figure 1.1: The NACHI MR20 7-axis Robotic Arm

Olimex Interface

The Olimex interface is a Linux single board that is connected between the NACHI AX20s main- and servo board. The Olimex intercepts the signals and sends out new manipulated signals depending on the interpretations provided by code running on the computer. The software running on the Olimex is written by Johannes Schrimpf.

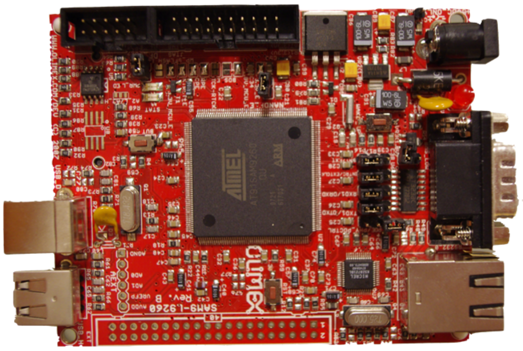


Figure 1.2: The Olimex interface.

Gripper
The Schunk PG-70, which is an electrical 2-finger parallel gripper, was used for the thesis. It was connected to the server computer using a CAN-USB interface by esd - electronic system design.

Figure 1.3: The Schunk gripper

Microsoft Kinect
The Microsoft Kinect is a motion sensing device, and is based on range camera technology by PrimeSense (PrimeSense, 2012). It comprises of an infrared projector, an IR camera and a RGB camera. The Kinect is capable of providing depth maps within the range of 0.7m-6m.

Figure 1.4: Microsoft Kinect.

Arduino Uno
The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins, 6 analog inputs and a 16 MHz crystal oscillator.

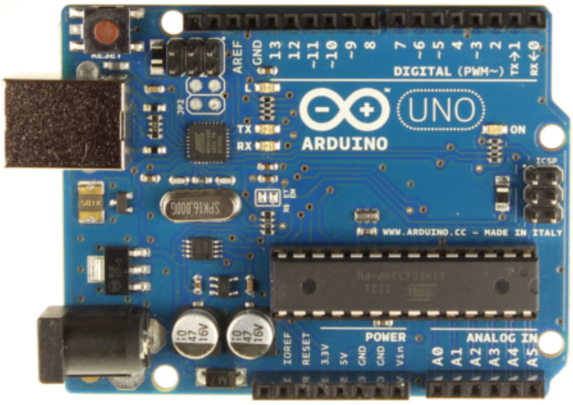


Figure 1.5: Arduino Uno.

9DOF Sensor Stick

The 9DOF Sensor stick by SparkFun is a small sensor board with triaxial accelerometers, magnetometers and gyroscopes. Through an I2C interface, one can access the following sensors:

ADXL345 - triaxial accelerometer
HMC5883L - triaxial magnetometer
ITG-3200 - triaxial gyroscope

The 9DOF Sensor Stick is referred to as the MARG-sensor (Magnetic, Angular Rate, Gravity) in the thesis.

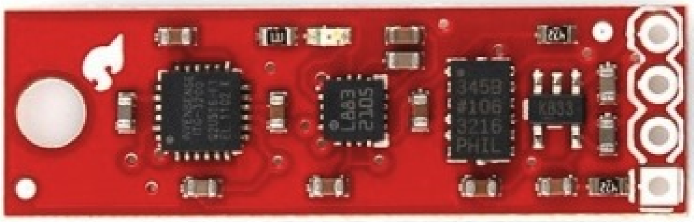


Figure 1.6: 9DOF Sensor Stick.

Cisco WVC210 Wireless-G (PTZ) Internet Video Camera

The Cisco WVC210 pan/tilt/zoom camera is an IP camera that supports two-way audio, and can provide both MPEG-4 and MJEG video streams. Its panning range is $\pm 67°$ and its tilt-range is $-34° \rightarrow 30°$.

Figure 1.7: Cisco WVC210 Wireless-G (PTZ) Internet Video Camera

Computers
A workstation and server computer was available for use.

The workstation's specifications were: Core i5-2500K processor clocked at 3.3GHz (4cores) with 6GB memory and a AMD Radeon HD 6900 graphics card. It was also accompanied by 3 x Samsung SyncMaster SA550 27" screens.

The server computer was a Dell Dimension 5000, with the following specifications: Intel Pentium 4 3GHz (2cores) with 1024MB ram and integrated graphics.

1.4 Available Software

There were no limitations on what software that could be used. However, a LabVIEW framework for communication with some of the relevant hardware was made available by Balazs Daniel. Therefore it was chosen to utilize LabVIEW 2011 with National Instruments' Robotics Module for all code on the server side.

In addition, various software pertaining to an existing remote operation system, CROP (Cognitive Remote Operation System), was available for repurposing. Aspects regarding the use of these components will become apparent later in the thesis.

1.5 Limitations

The problem description was handed out January 16th, and the master thesis was due June 11th, which gives a total of 100 working days.

Considering the time constraints and the main goal of developing and implementing a system for remote operation of industrial robots, extensive theoretical analysis into certain topics is considered to be beyond the scope of this master thesis.

For outlining an example of this: While the Microsoft Kinect is used in the practical implementation, the focus is not placed on the theoretical foundations of object tracking algorithms. Instead, the focus is on the utilization of the tracking data in the remote operation context.

This also applies to the implementation of inertial-based tracking.

1.6 Structure of the Report

The remainder of the report is structured as follows.

Chapter 2 gives an introduction to remote operation. The chapter holds a brief history of teleoperation, an introduction to the applications of industrial robots, the challenges of remote operation, and lastly a brief introduction to CogInfoCom and its terminology.

Chapter 3 presents programming and control in remote operation. It contains an overview of robot programming methods as well as 6DOF[1] control devices.

Chapter 4 explores communication during remote operation. The challenges of handling and structuring information is discussed. Lastly, some considerations regarding data transfer rates between the remote operation system and the physical robot system.

Chapter 5 gives a high-level system description of the proposed cognitive remote operation system. The purpose of this chapter is to provide an overview of the various system functions and behavior.

Chapter 6 describes the practical implementation by dividing it into the remote room and the local room.

Chapter 7 aims to provide a user guide with all the necessary information to set up and run the developed cognitive remote operation system.

Chapter 8 presents the experiments performed for verifying the various system functions, as well as usability tests performed with 5 participants.

Chapter 9 holds discussions regarding the topics of control and programming as well as multimodal communication in light of the implementations and experiments.

Chapter 10 will conclude the report through a conclusion and suggestions of topics for further work.

Digital Attachment contains videos of the experiments and the code developed for the thesis. See Appendix C for an overview of the contents in the digital attachment.

---

[1] 6DOF - 6 Degrees of freedom

Chapter 2

Introduction to Remote Operation

This chapter aims to give a brief introduction to *Remote Operation of Industrial Robots* and the challenges within the field. Remote operation shares much of its theoretical foundations with *Teleoperation*. The main differences between remote operation of industrial robots and teleoperation can be attributed to the need it it is supposed to fulfill. Teleoperation is intended to be an extension of the human to perform operations. The aim of remote operation of industrial robots, however, is to support in changeover and monitoring of production tasks.

2.1 Teleoperation

As teleoperation is considered a big part of the theoretical foundations of remote operation, this section aims to give a brief introduction to its history and evolution.

Most historical details in this section are taken from Bejczy (2001), Hendseth (1994) and Niemeyer et al. (2008).

Teleoperation has a history that dates back to the 1940s and 1950s. As one became increasingly more conscious of the dangers of nuclear radiation, a need for telemanipulation arose. Raymond C. Goertz created systems for humans to handle nuclear material. From behind shielded 1m thick walls, the operator could handle the material with natural hand motions.

Figure 2.1: Raymond C. Goertz teleoperating in the early 1950s (Niemeyer et al., 2008)

He developed a mechanical system comprised of a series of gears, linkages and cables which allowed the operator to feel reaction forces and vibrations through the connecting structure. The systems were master-slave systems, meaning that both the operator's controller and the manipulator were kinematically identical. This system consequently provided full one-to-one correspondence between the two, in terms of both joint- and spatial movement.

Goertz' first mechanical master-slave system had limitations in terms of the distance between the operator and the remote location. While it was sufficient for the initial trails at relatively short distances, where the operator observed the scene through protected viewing ports in the shielded walls, Goertz recognized the value of electrically coupled manipulators. In the process of shifting focus from mechanical linkages as the transmission-medium over to electric signals, he laid the foundations of modern teleoperation and bilateral force-reflecting positional servos. Figure 2.2 shows the components of such a system.

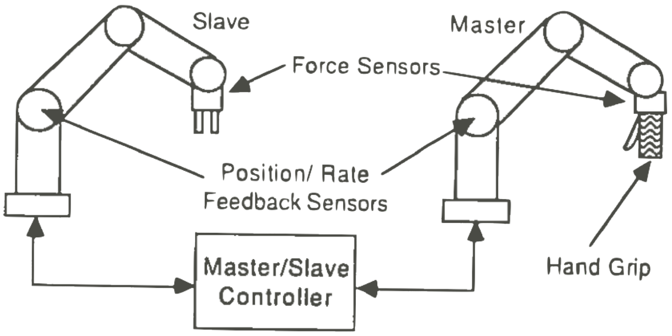


Figure 2.2: Components of a master-slave controller (Karwowski and Rahimi, 1992)

Bejczy and Handlykken (1981) generalized bilateral control in teleoperation, and developed what they called the *Universal force-reflecting hand controller* (UFRHC). It was a mechanical controller providing 6DOF control without the need for geometrical equality between the master and the slave. Whereas a traditional one-to-one master-slave would be directly reflected in joint-space, the UFRHC was based on the controller's position/orientation (hereby referred to as pose) by geometrical transformations, allowing the master and slave to be kinematically dissimilar. The first incarnation of such a controller, with a work volume of 30cm^3, was developed by Antal K. Bejczy at the Jet Propulsion Lab, and is depicted in figure 2.3.

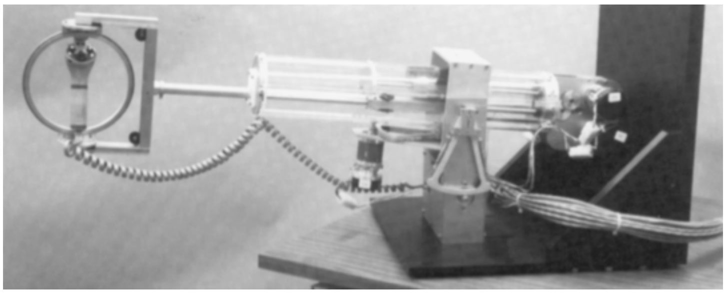


Figure 2.3: NASA JPL's implementation of Universal Force-Reflecting Hand Controller (Brooks and Bejczy, 1985).

Since the early work of Goertz in bilateral force-reflecting positional servos, the master and slave has been separated by increasingly longer distances. Early examples of this are the MANTIS 1 & 2 systems, which were used for maintenance related tasks in ’hot’ areas of the CERN accelerator tunnels. The system’s slave-arms were placed on hydraulic cranes onboard 4-wheeled drive base vehicles, and connected to the master system by long self-deploying umbilical cables (Horne, 1978; Horne et al., 1987). A picture of a MANTIS vehicle for the slave can be seen in figure 2.4.

Figure 2.4: MANTIS vehicle (CERN Document Server).

Teleoperation has since expanded from being used only for work in radioactive areas. In 1993, the first telerobotic system went to space on board the Columbia space shuttle. Furthermore, in 2001, a surgeon used a telerobotic system for performing the first transatlantic surgery from New York to France.

Today, teleoperation has found its place in the military, space- and subsea exploration and hospital operating rooms as well as in other fields. As pointed to by Whitney (1985), it also spawned the field of force control. In contrast to teleoperation, force controlled robots rely on computers rather than humans to execute control strategies, thus demoting humans from being an integral part of the control-loop.

2.2 Applications of Industrial Robots

Remote operation systems are intended for support in programming, changeover and monitoring of industrial robot systems. In light of this, a brief overview of the use of industrial robots will be given.

The data figure 2.5 is based on the production and shipments of manipulators and robots by members of Japan Robot Association in 2011.

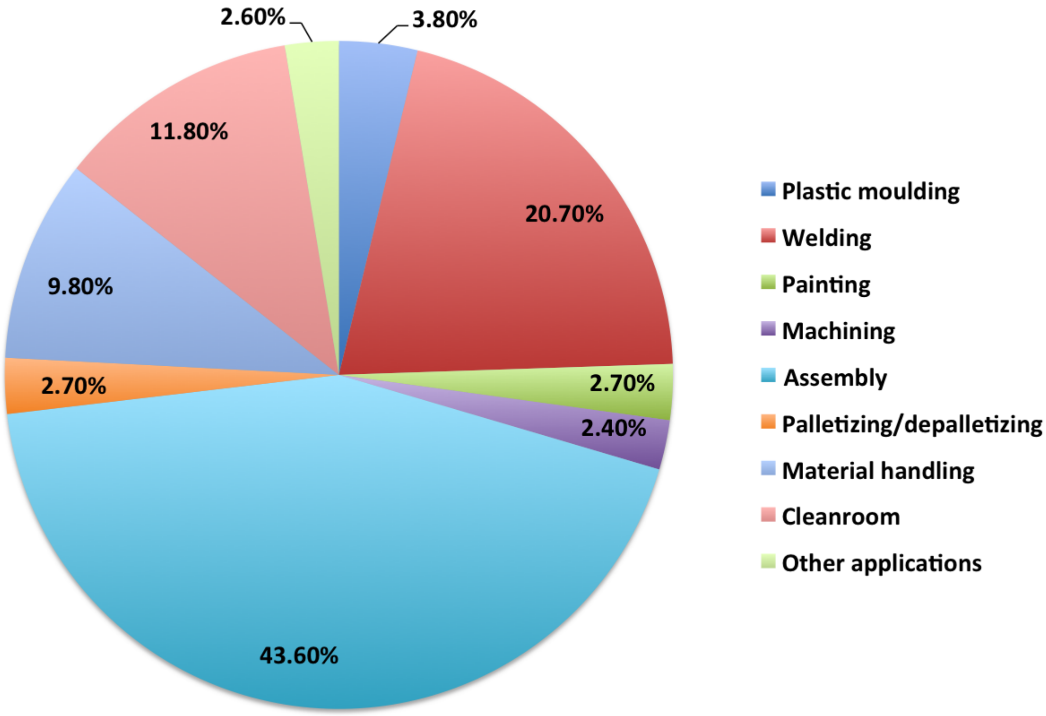


Figure 2.5: Areas of application for robots sold in Japan 2011 (JARA, 2012)

The chart shows that the diversity of applications are considerable. This section briefly present some of these.

2.2.1 Material handling

In material handling, the robot's basic capability to transport objects are utilized. In material handling, industrial robots may provide (Nof, 1999):

- Shorter cycle times
- Reduction of direct labour costs
- Less damage to parts during handling
- Reduction of tedious and fatiguing tasks for personnel

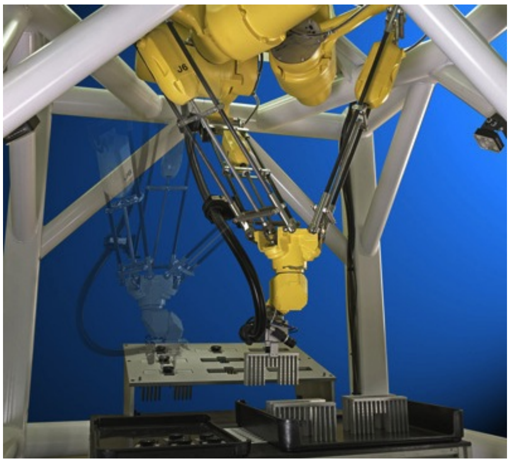


Figure 2.6: Small part handling (Fanucrobotics.com, 2012).

Figure 2.7: Palletizing (Motoman.com, 2012).

2.2.2 Machine Tending

Machine tending is where an industrial robot is used for loading and unloading workpieces in various machining processes. In machine tending, industrial robots may provide (Nof, 1999):

- Shorter cycle times
- Increased production uptime
- Removing personnel from monotonous tasks, or from dangerous environments

Figure 2.8: MR20 tending hobbing machine (Nachirobotics.com, 2012).

Figure 2.9: KUKA robot in foundry operation (Kuka-robotics.com, 2012).

2.2.3 Painting

Exposing humans to paint over longer time spans can cause damage to the nervous system. Therefore, productions with spraying/painting processes can utilize industrial robots in order to remove humans for such toxic environments. In addition, the use of industrial robots can give higher consistency in the painting process, as well as reduced waste of coating materials (Nof, 1999).

Figure 2.10: Coating rotor blades (ABB.com, 2012).

Figure 2.11: Coating of car parts (Fanucrobotics.com, 2012).

2.2.4 Welding

Due to the harsh environment filled with fumes and blinding light, humans need frequent breaks when welding. Industrial robots can therefore offer considerable increases in actual welding time. Due to the good repeatability of a robot, higher product quality can often be observed.

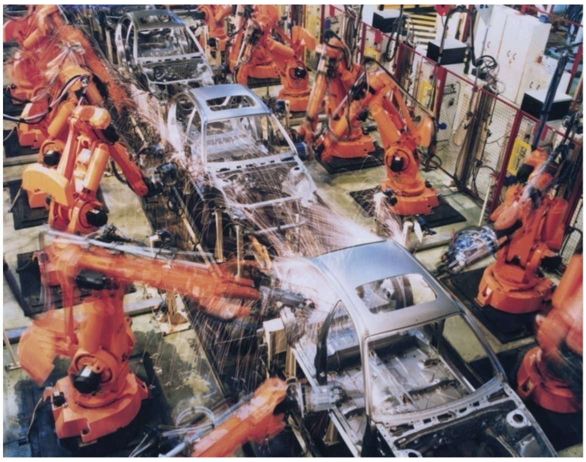


Figure 2.12: Welding operation of Fiat (ABB.com, 2012).

Figure 2.13: Arc welding operation (Nachirobotics.com, 2012).

2.2.5 Machining

In machining applications, the robot typically holds a powered spindle, and performs tasks such as grinding, de-burring, drilling or milling. Dedicated machining centers have very stiff constructions to accommodate high forces. Industrial robots are not as stiff, which can cause dynamics that are difficult to model and control. Consequently, industrial robots can not be used in all cases, and does not provide as high quality as dedicated machining centers, while they can be sufficient for many processes (Nof, 1999).

Figure 2.14: Grinding (Fanucrobotics.com, 2012).

Figure 2.15: De-burring (Motoman.com, 2012).

2.2.6 Assembly

Assembly operations are often very advanced. Through the use of technologies such as vision systems, tactile sensors, tool-changing stations etc., industrial robots can be used for increasingly more advanced assembly tasks.

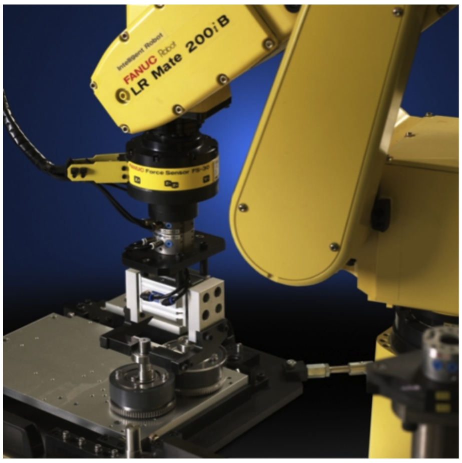


Figure 2.16: Fanuc during assembly operation (Fanucrobotics.com, 2012).

Figure 2.17: Robots cooperating for assembly (Fanucrobotics.com, 2012).

2.3 Challenges of Remote Operation

This section aims to present some of the challenges present in remote operation of industrial robots. The focus will be on challenges related to sensory feedback and dexterity.

In everyday life, as humans navigate and interact with their environment, they can take advantage of all their trained senses. By using a combination of stereo- and peripheral vision, sound, touch and kinesthetics, humans are able to skillfully interact with objects and perform tasks with precision. The integration and classification of sensory information into meaningful information is what we call perception, and this is an integral part of human behaviors, such as selective attention[1] and orienting reflexes[2] (Atkinson et al., 1999; Sokolov, 1962). In local operation, these perceptual concepts are a part of what makes a trained operator adept in manipulating robots, while handily avoiding singularities and obstacles. In remote operation, the operator is often deprived of some of the sensory information otherwise naturally acquired, consequently making control of the robot strenuous for the operator (Thomessen et al., 2011; Basanez and Suarez, 2009; Thomessen and Kosicki, 2011).

For addressing the challenge of supplying the operator with the necessary information to control robots, multi-modal systems were developed. A multi-modal human interface uses the previously mentioned modalities inherent in humans' perceptual system, such as vision, audition, sound etc. for serving two primary functions (Aracil et al., 2007):

- Capturing the operator's intension for commanding the system.
- Stimulate the operator's senses with information from the remote environment.

As multi-modal systems relate to humans' senses, the systems have to be designed to avoid unintuitive responses in the utilized perceptual modalities. An examples of an unintuitive response due to discrepancies between the kinesthetically delivered command and the expected visual response can be seen in figure 2.18.

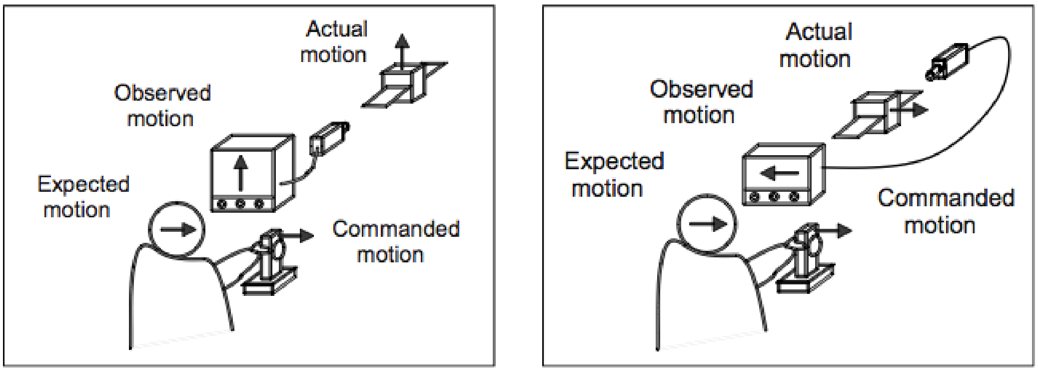


Figure 2.18: Visual disadjustment due to kinematic reasons (left) and observational reasons (right)(Aracil et al., 2007).

---

[1] The ability to select some information for more detailed inspection, while ignoring other information (direct excerpt from Atkinson et al. (1999)).

[2] Involuntary body movement that orient the receptor in question towards the part of the environment causing the receptors stimulus' change (Thomessen and Kosicki, 2011).

In multi-modal systems, information based on physical phenomenons imperceptible by humans can also be conveyed. Thus, information such as magnetic field readings, ultra-sonic readings etc., can be delivered to the human operator. In contrast to the example in figure 2.18, the use of information types that are incompatible with humans, may not have a self-evident modality for delivery to a human. Consequently, the information has to be bridged in order to be compatible with the humans' cognitive system. CogInfoCom (Cognitive Info-Communications), is a research field exploring communication of information towards humans' cognitive systems. CogInfoCom uses the cognitive sciences as a part of its theoretical foundation, and can be used as a tool for efficient info communication in such situations. CogInfoCom will be further investigated in section 2.4. (Baranyi and Csapo, 2010)

Remote operation and teleoperation share the challenges of unstructured remote environments, communication delays, human operator uncertainty and safety. In teleoperation, what is called *teleoperational aids*, are aimed at assisting in overcoming these challenges. Artificial fixtures and motion guidance are among these, and can be divided into two (Basanez and Suarez, 2009):

Software-based Geometric constraints added in software for constraining the operator's motions based on points, lines, planes, spheres and cylinders.

Hardware-based Physical hardware such as guide rails, sliders etc.

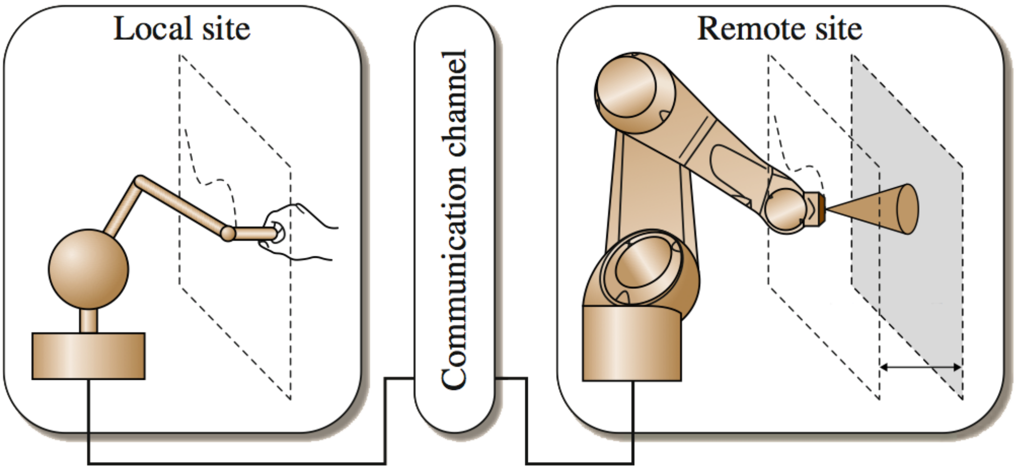


Figure 2.19: Illustration of painting operation with plane constraints (based on Basanez and Suarez (2009)).

Such constraints can give the operator increased confidence during the remote operation, and may offer safety in terms of avoiding collision with objects in the remote environment. One of the advantages of software-based constraints, is that they can easily be changed and modified during operation.

Force-reflection has been available in teleoperation from its early days, as presented in section 2.1. It offers an important feedback from the remote manipulator in its natural modality and has significant improvements to dexterity in teleoperation (Horne, 1988).

However, the negative effects of time-delay in such systems became a topic of discussion already in the 1960s (Niemeyer et al., 2008). Over longer distances, thus high delay, force-reflection becomes increasingly difficult to perform without resulting in resonance-effects between the human operator and the system, consequently making the system unstable. Some implementations for long-distance force-reflection, uses virtual models of the robot's environment to create virtual predictive models, consequently circumventing the strict requirements for low latency. This approach, however, puts very high requirements on the accuracy of the virtual models, which can be time-consuming and costly to establish (Burdea and Member, 1999). Due to these difficulties, force-reflection is often omitted in long-distance systems.

2.4 CogInfoCom

CogInfoCom (Cognitive Infocommunications) is a multidisciplinary field which aims to investigate the link between the research areas of infocommunications and the cognitive sciences. Baranyi and Csapo (2010) pointed to a trend of overlapping, where engineering systems, based on synergies between these research areas, emerged. CogInfoCom was created as a response to this convergence, and aims to provide a systematic view of how cognitive processes can co-evolve with info communication devices.

In developing a remote operation system for industrial robots, it is important to have a holistic view. This means that all aspects related to both the operator's cognitive system, the robot, and how to most efficiently communicate between these, must be taken into consideration. CogInfoCom can be seen as the intersection of this, as illustrated in figure 2.20, and can be used as an aid during system development.

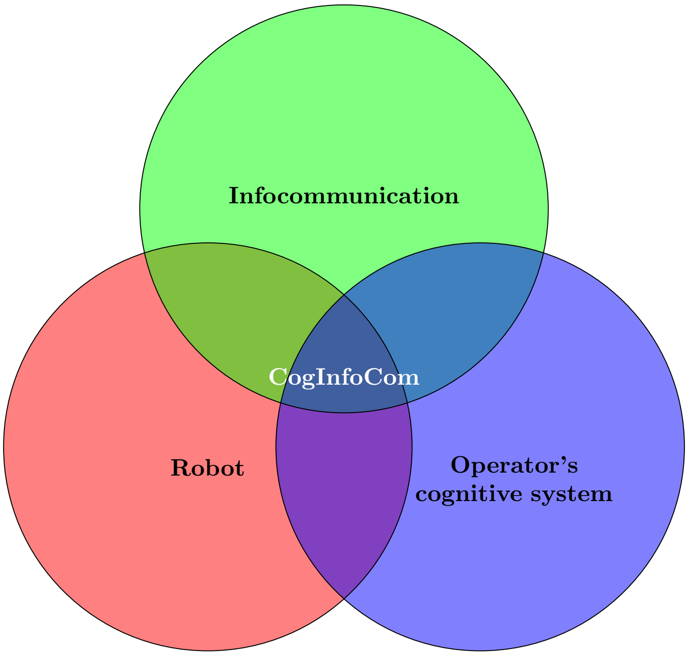


Figure 2.20: Illustration of how CogInfoCom can unite infocommunication, cognitive sciences and robotics.

A framework for discussing CogInfoCom systems has been defined. Most details pertaining to the definitions are taken from Baranyi et al. (2011); CogInfoCom.hu (2012); Csapo and Baranyi (2011). In CogInfoCom, it is distinguished between the *mode* and *type* of communication.

Mode refers to the actors at the ends of the communication. Intra-cognitive communication is between two cognitive beings of equal level of cognitive ability, while inter-cognitive communication is the mode between two cognitive beings of different capabilities (e.g. human and artificially cognitive system).

The **type** of communication, refers to the type of conveyed information and how it is performed:

Sensor-sharing communication Entities on both ends use the same sensory modality to perceive the communicated information.

Sensor-bridging communication - The sensory information obtained or experienced by the entities are not only transmitted, but transformed to an appropriate and different sensory modality.

Representation-sharing communication - The same representation of the information is used in both endpoints to communicate the information.

Representation-bridging communication - The sensory information transferred is adapted so that a different information representation is used at both ends.

A popular example of inter-cognitive sensor-bridging communications in modern cars, is the reversing collision alarms, where a frequency-modulated sound communicates the distance between the car and objects behind it. In the context of industrial robotics, an example of inter-cognitive sensor-sharing communication can be direct video transfer of the robot cell to the remote operator. This can give the operator information regarding its pose, nearing singularities, collisions and more. An example of inter-cognitive sensor-bridging communication in robotics, may be presenting forces or joint angles through auditory or visual modalities to the operator. This can tell the operator whether or not the end-effector is in contact with a workpiece, or if the robot is nearing a singular configuration.

The individual techniques and concepts in CogInfoCom are not new and have been employed in engineering systems for many years. The novelty lies in that infocommunication is viewed as an integrated process influenced by the cognitive capabilities of both sides (Baranyi and Csapo, 2010).

Some of the research utilizing the CogInfoCom in robotics, has been related to emotional and social human-robot interaction (Buss et al., 2011; Han et al., 2011) and force feedback in grasping for virtual reality environments and telemanipulation (Galambos and Baranyi, 2011).

In terms of applications directly aimed towards remote operation of industrial robots, there have been research in utilizing visual and auditory feedback from the physical robot system to enable efficient operation. Implemented examples includes earcons[3], such as the well known USB insert/remove sound to communicate whether contact between the end-effector and surface was established. Also other examples such as auditory frequency-modulated feedback based on approaching singularities have been tested (Thomessen et al., 2011).

---

[3] Auditory icon providing immediate and unique semantic interpretation

Chapter 3

Programming and Control in Remote Operation

For a remote operation system that allows system integrators to provide remote support in changeover for SMEs, the methods for programming and control can significantly influence the effectiveness of the process. In light of this, the upcoming sections give an introduction to robot programming and 6DOF control devices.

3.1 Robot Programming

A prerequisite for a robot perform a task, is for it to be programmed to do so. There are many methods for programming robots, and this section aims to shed light on some of these.

Company costs associated with programming robots can be considerable, as it often requires time by trained personnel. This cost is even higher if the enterprise has to bring external talent to the location for the programming. An example of how much time and effort that goes into programming industrial robots, can be seen when programming for grinding applications. In grinding applications, the programming time is roughly 400 times the execution time of the grinding sequence (Thomessen and Lien, 2004).

In Biggs and Macdonald (2003)'s survey of robot programming systems, there is a division into manual- and automatic programming methods. *Manual programming* is what they used as a collective term for programming where the user has full control of the robot code, meaning that all program lines and blocks are made manually. In contrast, they use *automatic programming* as a term for methods where the programmer has little to no direct control over the creation of the robot code. This means that the program lines and blocks are made for him automatically based on instructions from tools like e.g. a teach-pendant or other modes of input. Automatic programming is mostly done on-line, either with the real robot with all its servos running, or a full virtual model of the robot is simulated and used for reference in the programming.

The workflow of combining automatic and manual programming is often referred to as *hybrid programming*. In hybrid programming, the resulting program from an automatic programming method is modified and adjusted by manual programming.

For this thesis, manual programming is considered out of the scope of this work, while automatic programming will be further elaborated on.

Biggs and Macdonald (2003) further divides automatic programming into *Instructive Systems*, *Learning Systems* and *Programming by demonstration*.

In programming using instructive systems, the robot is supplied with a sequence of instructions. The instructions often hold commands for the robot to perform full subtasks, and is in many situations supplied in real-time. These subtasks, however, haveto be programmed or trained in advance.

Learning systems often employ computational intelligence methods such as neural networks and fuzzy logic for creating programs by inference based on self-exploration or provided examples.

Programming by demonstration is where the robot learns by imitating the result of the programmers input to the system. The input can be a continuous set of locations or joint angles for creating full line-segments by using e.g. lead-by-the-nose[1]. While Biggs and Macdonald (2003) include point-by-point learning using a teach-pendant as a part of programming by demonstration, this process is often distinguished by others and called *teach-in programming* (Prof. Trygve Thomessen). In teach-in programming, a typical workflow for the operator is to move the robot to points, pertinent for performing the task in question, and recording these to create the program. This is one of the most common methods for programming.

From the view of a remote operator and the problem presented in this thesis, instructive methods could be considered too high-level. However, programming by demonstration and teach-in programming are considered by the candidate to provide the necessary functionality for solving the problem, while some aspects of learning systems could be considered for possible future expansion.

---

[1] The robot servos are either made limb, or force control is used for allowing the operator to move the robot through space for demonstrating the movement.

3.2 6DOF Control Devices

This section aims to provide a brief overview of various control devices enabling 6DOF control of industrial robots. The focus will be on control devices that can be considered practical in providing one-to-one pose. Joint-space reflective solution such as the traditional one-to-one master-slave or replica controllers of teleoperation will not be considered. The reasoning for this is as follows: Remote operation as a support system for robot integrators, has a greater benefit of having a control method adaptable to all industrial robot systems, regardless of kinematical configuration. Thus, control devices based on acquiring a reference pose rather than joint-angles are indiscriminate towards the kinematical configuration of the controlled robot.

In presenting the control devices, it will be differentiated between mechanical- or non-mechanical based capture systems.

3.2.1 Mechanical Capture Systems

As seen in section 2.1, mechanical means for acquiring one-to-one 6DOF control were popular in the early days of teleoperation. In this type of capture systems, mechanical linkages connects a *reference* to a *target* as depicted in figure 3.1. By forward kinematics, the targets pose relative to the reference can be determined. The joint angles are typically measured using incremental- or absolute encoders or electromechanical potentiometers (Rolland et al., 2001).

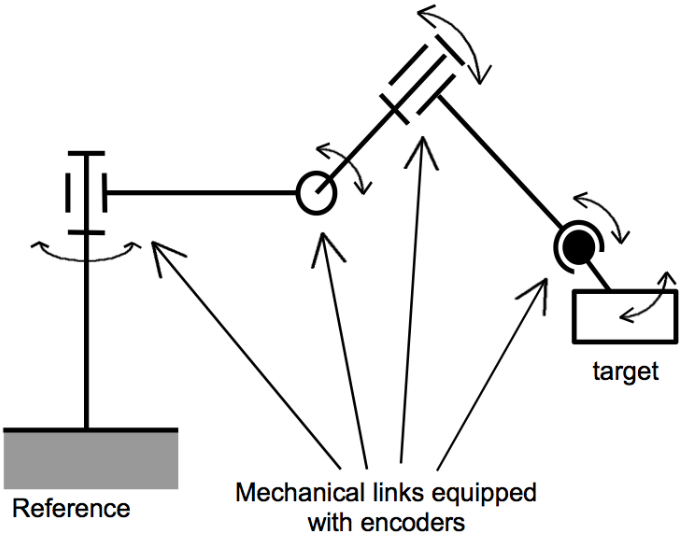


Figure 3.1: Typical structure of a mechanically linked tracking system (Rolland et al., 2001).

Examples of mechanical linkage based tracking systems are measuring arms, such as the one depicted in figure 3.2. Thomessen and Lien (2004) presented approaches for intuitive robot programming using such devices for reducing programming- and training time. Mechanical-based capture systems also allow forces to be backdriven into the controller, such as the UFRHC presented in section 2.1.

Figure 3.2: ROMER measuring arm (Romer.eu, 2012).

3.2.2 Non-mechanical Capture Systems

Brooks and Bejczy (1985) did a survey of various input devices for control of 6DOF manipulators. In this survey, what was dubbed as *the universal floating-handle controller*, was among the presented. It is a 6DOF control device, which does not determine its pose using joints or linkages. One example used for this control was the data-glove, which allowed for full pose information without mechanical tracking techniques.

There are several capture systems that can be used for realizing a universal floating-handle controller. They can be based on techniques such as magnetic-, acoustic- , vision- and inertial tracking (Rolland et al., 2001). In the interest of brevity, magnetic-, vision- and inertial based tracking will be further described.

Magnetic field-based tracking
In most magnetic tracking techniques, the *transmitter*, consists of three orthogonal coils, circulated with electrical current to produce electromagnetic fields. A receiver, usually consisting of three orthogonal magnetic field sensor, is placed in the electromagnetic fields' vicinity. The flux induced by the electromagnetic field in a given sensor, is a function of its distance and angle relative to the coils. Thus, the sensor output is used to establish the receivers position and orientation in space (Raab et al., 1979; Rolland et al., 2001).

There has also been approaches for reducing the complexity and footprint of magnetic tracking systems by using biaxial receivers or transmitters (Paperno et al., 2001; Paperno and Keisar, 2004).

Figure 3.3: Magnetic tracking system by Ascension with transmitter on the left and four receivers on the right (Ascension-tech.com, 2012).

Vision-based tracking
There is a plethora of vision and optical based tracking techniques. Some are based on using multiple cameras for performing triangulation of features or markers (either passive or active) to determine the object's pose. There are also solutions where it is sufficient with a single CMOS camera, such as 3D-model based tracking. It uses a reference 3D-model of the object to be tracked. By tracking markers, features or silhouettes and comparing it to the reference model, the objects pose can be determined. In figure 3.4, a screenshot from such a solution can be seen, where the 3D-models wireframe is superimposed over the tracked object.(Rasmussen et al., 2001; Lepetit and Fua, 2005; Rolland et al., 2001; Moeslund and Granum, 2001)

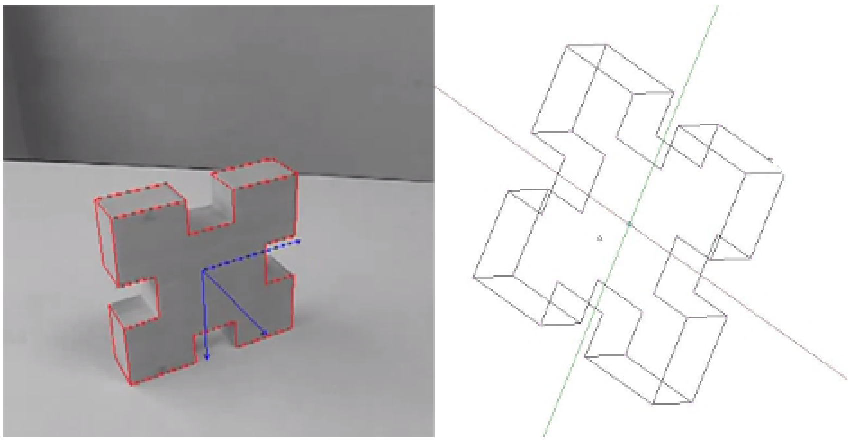


Figure 3.4: 3D-model based tracking using Visual Servoing Platform (ViSP, 2012).

In the case of the Microsoft Kinect sensor, a range imaging solution based on projecting a coded pattern of infrared dots into the room is used. Using an infrared camera, the deformation of the projected pattern is determined and used for establishing a depth map (PrimeSense, 2012).

Inertial-based tracking

Inertial-based tracking utilizes an *Inertial Measuring Unit* (IMU), which usually holds a 3-axis accelerometer and a 3-axis gyroscope. It can be used for determining full pose. However, low-cost consumer-grade IMUs are mostly used for determining the orientation alone, due to accumulation of positional error.

The basic idea for establishing the relative position using IMUs, is by integrating acceleration twice with respect to time. The integrated acceleration has to be gravity-compensated, as the accelerometers measure the gravity and any additional acceleration. Gravity compensation can be difficult if the system is in full motion due to dynamic forces. Misaligned gravity vectors is considered one of the biggest sources of error (Gagné, 2011).

Gyroscopes provide angular rates with respect to itself, meaning that relative orientation can be obtained by integrating once. Any errors in the readings will also be accumulated will cause drift. The absolute roll and pitch can be determined using the accelerometers, and therefore, many implementations utilize complementary filters using both gyro- and accelerometer data in determining the orientation. However, accelerometers cannot compensate for the gyroscopic drift in yaw, introducing the need for magnetometers. As the word suggests, magnetometers provide magnetic readings, and enables determination of the absolute yaw by referencing it against the earths magnetic field. When a 3-axis magnetometer is added to an IMU, it is often referred to as a MARG-sensor (Magnetic, Angular Rate, Gravity) (Madgwick et al., 2011).

While the methods presented in this section can provide similar data, the workflow in using them may differ somewhat. This is due to what is called *clutching* (also known as *indexing*) (Niemeyer et al., 2008). Clutching is related to the coupling of the robot and the controller. If the controller and the slave robot have the same initial position, no clutching is needed. However, if this is not the case, their positions have to be either synchronized or control has to be initialized and applied with an offset. This process of initializing control with offsets is called clutching. It and can be seen as analogous with the act of lifting a computer mouse from the mousepad and repositioning it while the cursor is stationary. Clutching is important when the work space of the controller and robot differs. While the one-to-one master-slave systems used in teleoperation usually avoid this, the need for clutching in the presented control methods is highly dependent on their implementation. Clutching can be considered convenient in many cases. In addition to allowing the operator to reposition the controller and mapping to a new work space, clutching it also gives the operator an opportunity to rest during remote operation of the robot (Niemeyer et al., 2008).

In terms of performance, Rolland et al. (2001) considers mechanical capture systems to provide some of the best results in accuracy, update rate, lag, and resolution. They do, however, have the problem of being prone to self-occlusions (in some locations, the control may get in its own way), which can be awkward for the operator (Karwowski and Rahimi, 1992). In addition, Lepetit and Fua (2005)'s survey of *monocular model-based 3D tracking*, points to the fact that mechanical tracking tethers the user to a limited working space, as a general disadvantage.

For magnetic based tracking, Rolland et al. (2001) points to the weakness of distortions of the magnetic field created by the tracking system. The distortions can be caused by e.g. metal in the environment, and can lead to large tracking errors. The attenuation in magnetic based tracking systems can not be increased indefinitely, due concerns of its effects on humans. This causes a limited working volume, which has been pointed to as a general weakness of magnetic based tracking by Lepetit and Fua (2005); Rolland et al. (2001).

A general concern in using vision-based tracking, is that the object of interest can be prone to occlusion from the camera's perspective. In addition vision-based tracking techniques are often sensitive to optical noise and variations in lighting, which can cause bad tracking. Due to the fact that the performance of vision-based tracking techniques is highly dependent on both the hardware and the tracking software, general statements regarding performance can be difficult. However, lag is often raised as an issue due to high computational requirements for some vision-based tracking algorithms, (Lepetit and Fua, 2005; Rolland et al., 2001; Moeslund and Granum, 2001).

To conclude, all approaches have their weaknesses. However, as a way of dealing with these weaknesses, Rolland et al. (2001) propose the use of hybrid systems to provide an optimal performance with sensible trade-offs. An example of a hybrid system can be combining inertial- and vision-based tracking.

Chapter 4

Communication In Remote Operation

One of the challenges in remote operation is to communicate meaningful information to the operator. While sensors can obtain enormous amounts of quantitative data, it is not necessarily meaningful or important to the operator in an unprocessed state. This chapter aims to provide a brief introduction to the challenges of handling information in remote industrial robot systems. The emphasis will be on categorization of information and considerations regarding specification of the transfer-channel between the remote operation system and the physical robot system. In this thesis, *information* is considered data that is correctly interpreted in its defined context.

Figure 4.1 shows a general system structure of a remote industrial robot system with its components pertinent for providing meaningful data to the remote operator. This figure will be further explained, and used as a basis for further discussions in this chapter.

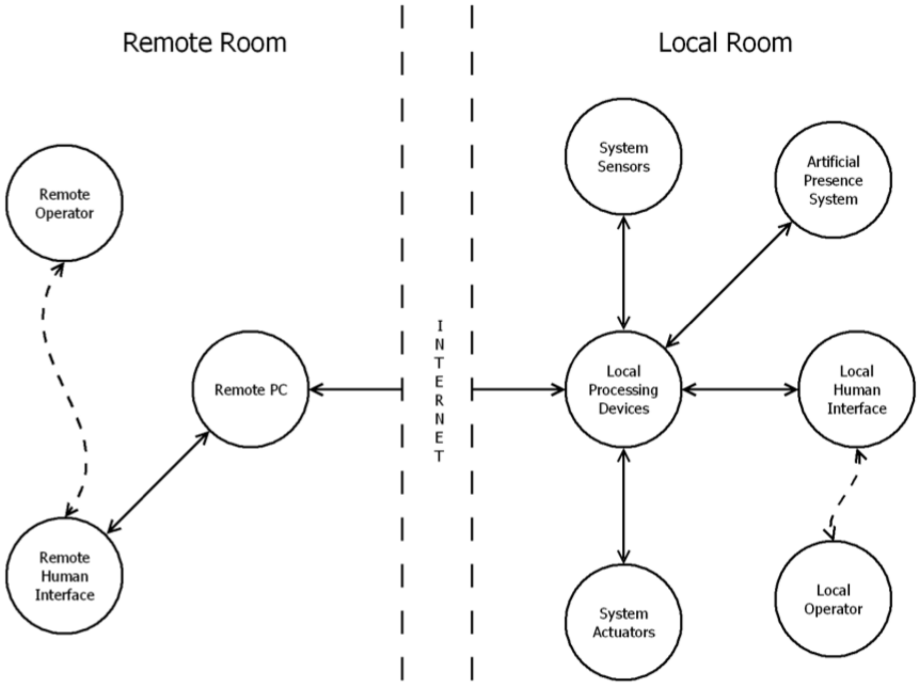


Figure 4.1: General system structure of a remote industrial robot system. (Thomessen et al., 2011)

In accordance with figure 4.1, the terms *local* and *remote* will be used in the remainder of the thesis, where:
Local - The location of the physical industrial robot system.
Remote - A remote location where the remote operator is located.

4.1 Transformation of Information

In figure 4.1, the *Artificial presence system* is a measurement system that gathers so-called *Naturally Acquired Data*[1], which has to be artificially gathered and provided to the remote operator. The *System Sensors* are measuring instruments gathering data pertinent for evaluating the present state of the active components in the local industrial robot system. It includes the robot's internal sensors and external sensors such as range, proximity etc. Also used in the system are *System Actuators*. These are executive devices such as the robot itself, or any additional machining or processing devices used in the robot operation. Lastly, the local human interface is the man-machine interface for the local operator, such as a teach-pendant, keyboard/mouse, joystick etc.

All these components communicates with the local processing devices, that are devices receiving data from all the aforementioned systems. These are responsible for interfacing between all the systems and providing them, and the remote PC, with adaptively filtered data necessary for their function.

The main challenge of the components in the local room, is to take the wealth of data in the local room and transform it from quantitative data, to qualitative data, and down to symbolic data if possible. In essence, it is desired to reduce the amount of data as much as possible, while still retaining the information. For outlining an example of this, consider the following situation: A 3D scanner has been placed in the local room with the intention of looking for collisions. It's raw data (quantitative) could be e.g. dense point clouds, while qualitative information could be distances between objects etc. However, at its core, the information the 3D scanner was intended to provide, is whether or not a collision has occurred, which could be represented symbolically.

CogInfoCom's concepts of sensor- and representation-bridging communication is not limited to being a tool merely for efficient cognitive infocommunication. If the application of these concepts results in more compact data, they could also be beneficial in facing the challenges of limited internet speeds.

---

[1] a term used by Thomessen and Kosicki (2011) for data that would be naturally acquired by the human senses, e.g. when locally operating a robot.

4.2 Proposed Structure

For developing a structure of the transfer-channel between the remote operation system and the physical robot system, the chosen approach is to evaluate and categorize the information from a functional standpoint. With this as a starting point, it is reasonable to appoint information pertinent to safety for both humans and equipment as the most important functional category.

In addition, two more main categories, *Control* and *Service, Setup & Maintenance*, are introduced. In figure 4.2, these categories are presented in descending order of importance.

Figure 4.2: Three main categories of information in the transfer-channel, ordered by the proposed importance.

As illustrated in figure 4.2, the borders between each of these categories are blurred, meaning that some information that could be considered control information in one situation, could in another situation be considered information pertinent for safety. This applies to the borders between all three categories, which will become apparent in the explanations for each of the categories in this section.

Safety
The category of *safety* is, as previously stated, intended to hold all information regarding the safety of both humans and equipment during remote operation. Information such as emergency stop signals, collisions, personnel entering the cell etc., is core information that should always be a part of this category. However, depending on the task and its specifications, the category can also include task-specific safety information. An example of this could be abnormally or dangerously high contact-forces during a grinding operation.

Control
The *control* category of the transfer-channel is responsible for all information needed for efficient control of the remote industrial robot system. Thus, information such as robot control commands and monitoring for evaluating the state of the robot and its environment, are considered a part of the control category. Task and process specific control- and monitoring information is also to be incorporated. Examples for a grinding process could be controls for the grinding motor, the contact-forces, surface temperature etc.

Service, Setup & Maintenance
Lastly, *service, setup & maintenance*, is the lowest prioritized category. Much of the information under this category does not have stringent requirements for delivery speed, and in some cases could be on-demand delivered. Examples of information in this category could be the robot controller's constants settings, its robot programs, or diagnostic data from the robot or other equipment in the setup.

4.3 Considerations Regarding Data Transfer

In this section, specification of a transfer-channel between the remote operation system and the physical robot system, with respect to bandwidth, will be explored. The specification will be performed for a hypothetical system using rough estimations and calculations, and should therefore be regarded as such. It is also important to emphasize that actual specification of a transfer-channel will vary considerably based on its and the systems morphology. The analysis is, however, intended to provide some indications of the bandwidth requirement for such systems, as well as how it is distributed across the categories.

Fundamental assumptions
It is assumed that all information that can be reasonably transferred as simple text-strings, are so, and that they are encoded and sent as ASCII characters. Moveover, a rough estimate for the size of most ASCII characters is $\approx$ 8bits pr character, which will be used here. These assumption may give higher bandwidth requirements than necessary, resulting in a conservative estimation of the needed bandwidth for these portions of the transferred information.

A certain level of artificial cognition is assumed in both ends of the transfer-channel. This means that the data is reasonable processed and interpreted by processing devices and sent through the transfer-channel to provide information, rather than raw data, as discussed in section 4.1. The level of the systems ability for this, will become apparent based on the presented example-information in the further part of this section.

With regard to update rate, the calculations are done under the assumption that Safety-information is to be provided at $200Hz$, text based Control-information at $100Hz$ and Service, Setup and Maintenance at $2Hz$.

The terms *upload* and *download* speeds will be used from the perspective of the local room. Consequently, these have to be reversed for considering the bandwidth requirements from the remote room.

Lastly, it is assumed that all communication is to be done using a protocol with minimal overhead, like e.g. UDP[2], and with minimal packet loss.

---

[2] UDP - User Datagram Protocol

Safety Category

Under the assumption of some level of processing before sending, it is assumed that the system is capable to send minimal amounts of safety information in a continuous stream. This means that rather than continuously sending detailed information for all safety subcategories, it is concatenating[3] statuses from all subcategories into one string, which is continuously sent. This way, the operator can immediately make out what type of safety-concern that has occurred, and respond appropriately.

An example of a continuously streamed safety-string could be:

```
"i,i,i,i,i,i"
```

Where `i` is an integer for reporting the safety status of a subcategory.

This example string equals the following amount of bits:

$$11\,\text{characters} \cdot 8\frac{\text{bits}}{\text{character}} = 88\,\text{bits}$$

Upload:

For outlining an example of information that could be represented in this string for a system used for a grinding process, these integers could represent information such as; Robot failure modes, collision, personnel entering the cell, approaching singularity, abnormal vibrations, grinding motor overheating etc.

Download:

For receiving safety information from the remote operator, a similar string of equal amount of characters can be considered in the interest of reducing complexity in the calculation. It could hold information such as emergency stop of the robot, stop of grinder or trigger an alarm in the local room.

With this as a baseline for safety-information, the bandwidth needed with a bilateral update rate of 200Hz becomes:

	Bits	Bandwidth
Upload	88	18kbps
Download	88	18kbps

Table 4.1: Bandwidth usage in kilo bit per second (kbps) for the safety category.

---

[3] Joining strings of information together.

Control Category

The category of *Control* contains considerably more information. In the interest of brevity, a template-string will be used for giving a rough estimate for the text-based information.

Numeric values in this category often needs additional digits per value for more precision. Examples of this are for strings representing 6-dimensional **p**-vectors for control-commands, or for force-vectors, process-parameters, joint-angles etc.. It is assumed to be sufficient with 8 numeric characters pr value in these cases. An example of such a string could be a force/torque-vector like this:

$$\begin{array}{cccccc} \texttt{"12.02123,} & \texttt{230.2263,} & \texttt{1.102133,} & \texttt{315.0173,} & \texttt{13.92421,} & \texttt{92.32113"} \\ \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ F_x & F_y & F_z & M_x & M_y & M_z \end{array}$$

On the basis of this, by including separators, an estimate of 72bits pr 8-digit value will be used in the calculations.

Upload:

The following text-based control information is assumed to be sent from the local room to the remote room in a system used for grinding:

Description	Amount of information	Bits
Robot's position/orientation vector	Six 8-digit values	≈ 432
Robot's joint-angles	Seven 8-digit values	≈ 504
Force/Torque vector	Six 8-digit values	≈ 432
Misc. process information	Six 8-digit values	≈ 432
Total		1800

Table 4.2: Uploaded text-based information in the control category.

Additional information related to the *Artificial presence system* has to be taken into account. In this example, it is assumed that this system consists of three camera streams. A traditional transfer rate for a 640x480 camera at 24frames pr second with audio could be ≈ 512kbps, depending on compression etc., and is used in this calculation.

Download:
The following control information is assumed to be received from the remote room in a system used for grinding:

Description	Amount of information	Bits
Desired position/orientation or joint-angles	Six 8-digit values	≈ 432
Grinder control	Three 8-digit values	≈ 216
Selection matrix	Assumed 6 binary values	≈ 88
Desired reference forces/torques	Six 8-digit values	≈ 432
Total		1168

Table 4.3: Downloaded text-based information in the control category.

With a bilateral update rate of 100Hz for text-based information, the bandwidth for the control category becomes:

	Description	Bits	Bandwidth
Upload	Text-information	1800	180kbps
	Three Cameras		1536kbps
	Total		**1716kbps**
Download	Text-information	1168	117kbps
	Total		**117kbps**

Table 4.4: Bandwidth usage for the control category.

Service, Setup & Maintenance
This category holds considerable amounts of information to be transferred. However, as the assumption that an update rate of $2Hz$ is sufficient for this data, it's contributions to the total bandwidth usage is somewhat negligible. For outlining this, 8000bits of data would only amount to a transfer rate of 16kbps.

In addition, much of this data could be considered not time sensitive, and could be collected on demand rather than a part of a continuous stream. For the estimates, it is assumed that the downloaded information is negligible, and the rough estimate of 16kbps is sufficient for the uploaded information.

Total bandwidth usage

By combining the estimates from all categories, the total estimated bandwidth requirement for the hypothetical system amounts to:

	Category	Bandwidth
Upload	Safety	18kbps
	Control	1716kbps
	Service, Setup & Maintenance	16kbps
	Total	**1750kbps**
Download	Safety	18kbps
	Control	117kbps
	Total	**135kbps**

Table 4.5: Total bandwidth usage.

Based on these calculations, it can be seen that the control category dominates the upload bandwidth requirements. This is mostly attributed to the camera streams. The chart in figure 4.3, shows a pie chart of the upload bandwidth requirements divided into text-based information and the camera streams. This indicates the importance of modest use of this type of information if possible.

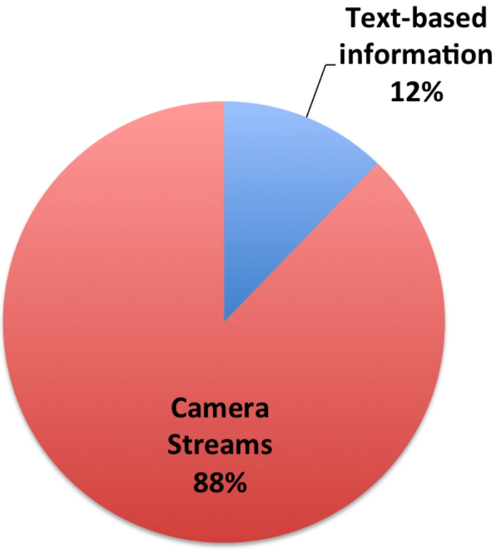


Figure 4.3: Pie chart of the upload bandwidth requirement divided into text-based information and camera streams.

Chapter 5

The Cognitive Remote Operation System

This section will describe the proposed remote operation system. The system utilizes approaches for control and programming presented in chapter 3, namely non-mechanical based tracking and teach-in programming. It also holds an approach for motion guidance, which was introduced in section 2.3. Lastly, it employs CogInfoCom for efficient communication of information to the remote operator.

5.1 Overview

The concept of the proposed programming- and control scheme can, at its highest system-levels, be described as follows. The remote operator moves a physical analog to the remote controlled robot's end-effector freely in space. The analog is equipped with a switch (hereby referred to as the *main switch*) for declaring that its movement, while the switch is activated, should be applied to the remote robot. The analog's change in pose relative to a fixed local coordinate system is acquired by non-mechanical based methods, sent to the remote location for processing, and applied in the robot's coordinate system. An illustration of an operator using the proposed system can be seen in figure 5.1. Also seen in the illustration, is the multiple screens available to the operator. The purpose of these are to convey crucial information regarding the control and programming of the robot, such as camera views of the remote location, active control modes, approaching singularities and more, which will be presented in section 5.4.

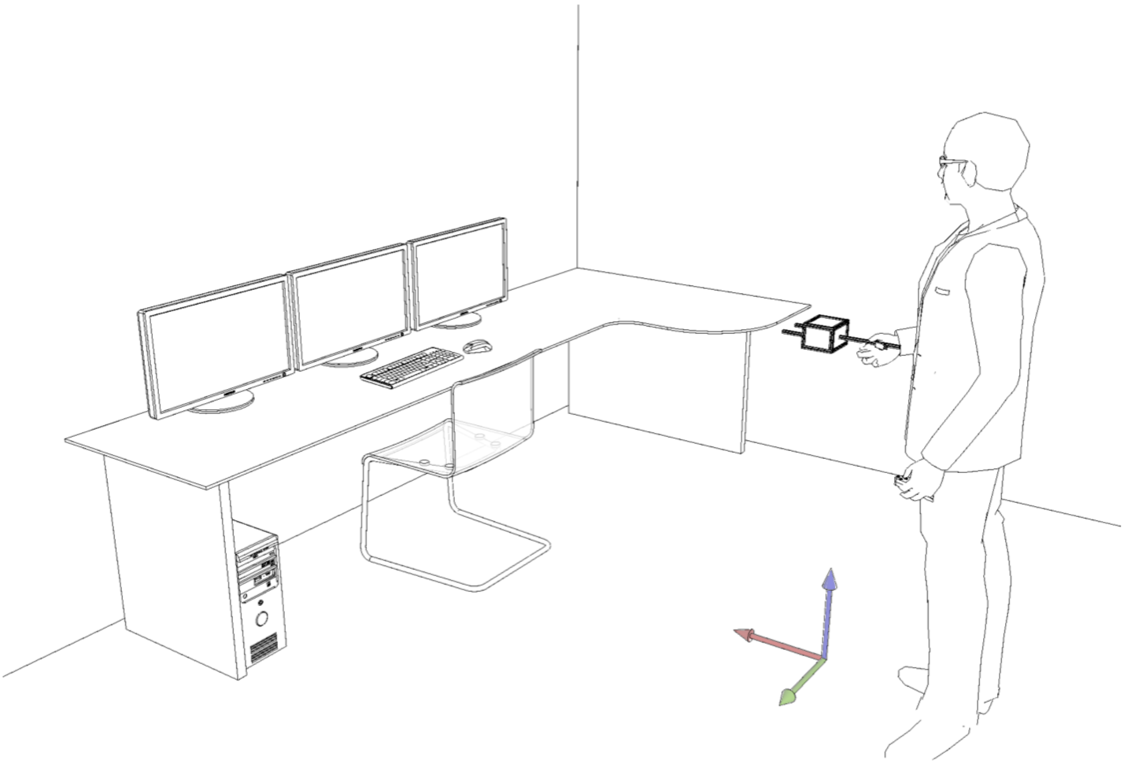


Figure 5.1: Illustration of an operator in the remote room with a gripper analog.

When using a hand-held physical object as the interface, it is referred to as a *Tangible Interface*. These interfaces aims to take advantage of the users everyday experience in manipulating familiar objects to give more natural and intuitive interfaces (Lepetit and Fua, 2005). The main reasoning for the use of a physical analog of the robot's end-effector as a 6DOF control device is two-fold:

1. It is assumed to give the operator understanding and insights into how his/hers actions translate to movement of the TCP[1] and how the required change in the robot's joint configuration to achieve this could turn out.
2. By adding geometrically similar obstacles in the remote room as in the local room, the fact that the dimensions of the analog gripper matches the real one's, could help the operator in interacting with, and navigating, the local room. Physical objects in the operator's environment could also be used as a way of adding constraints and motion guidance during remote operation.

The use of non-mechanical means for acquiring the pose of the analog, could provide a flexible solution with a relatively large workspace. This can be considered to reduce the amount of clutching during operation, which could be a inconvenient if the system is to be used for remote control of robots with large work volumes.

---

[1] TCP - Tool center point

In addition to the gripper analog itself, an *external controller* is to be used by the operator. It is intended to be placed in the operator's second hand, and is equipped with four buttons as illustrated i figure 5.2 &5.3.

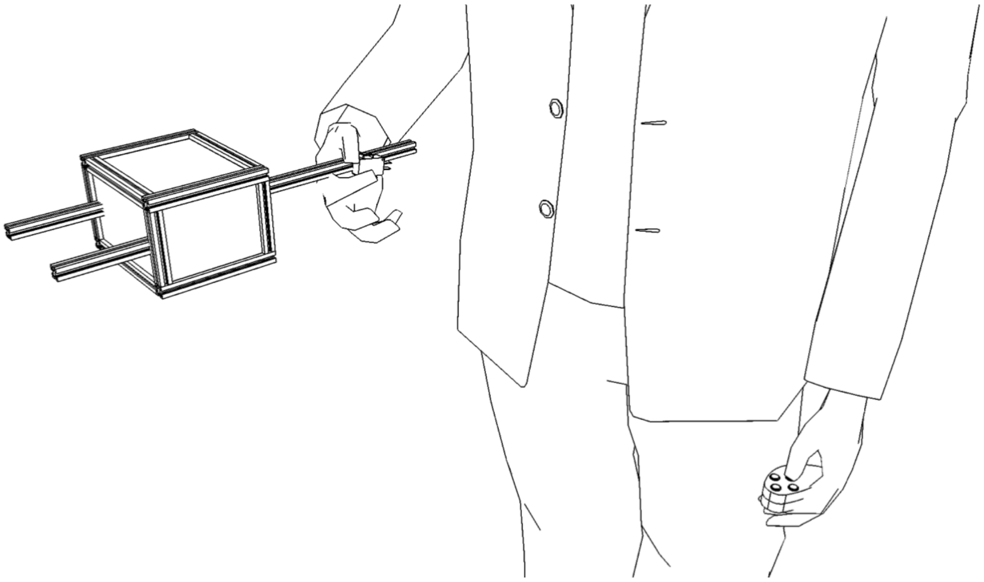


Figure 5.2: Illustration of an operator with the gripper analog in his right hand and the external controller in his left.

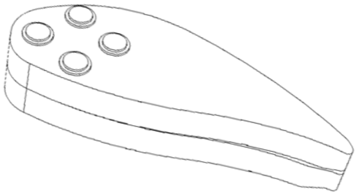


Figure 5.3: Illustration of the four-buttoned external controller.

The four buttons of the external controller gives the operator the ability to:

1. Quickly cycle through the control modes, which are; full 6DOF control, translation only, rotation only and reconfiguration (This will be further explained in section 5.3).
2. Quickly cycle through the presets for the control's scaling-setting's, 1:1, 1:2 and 1:3.
3. Instruct the robot to grip or release.
4. Save the robot's current pose to a robot program.

5.2 Coupling and Clutching

As presented in section 3.2.2, clutching is a process analogous with lifting and repositioning a computer mouse in order to reach a new area of the screen. In remote operation of industrial robots, clutching to an offset becomes a necessity in situations where the workspace between the controller and robot differs. The coordinate system which is used for reference, when control is activated, is called the coupling coordinate system.

Figure 5.4 illustrates the coordinate system which has been used for the coupling between the gripper analog and robot, each time control is initiated by pressing the main switch.

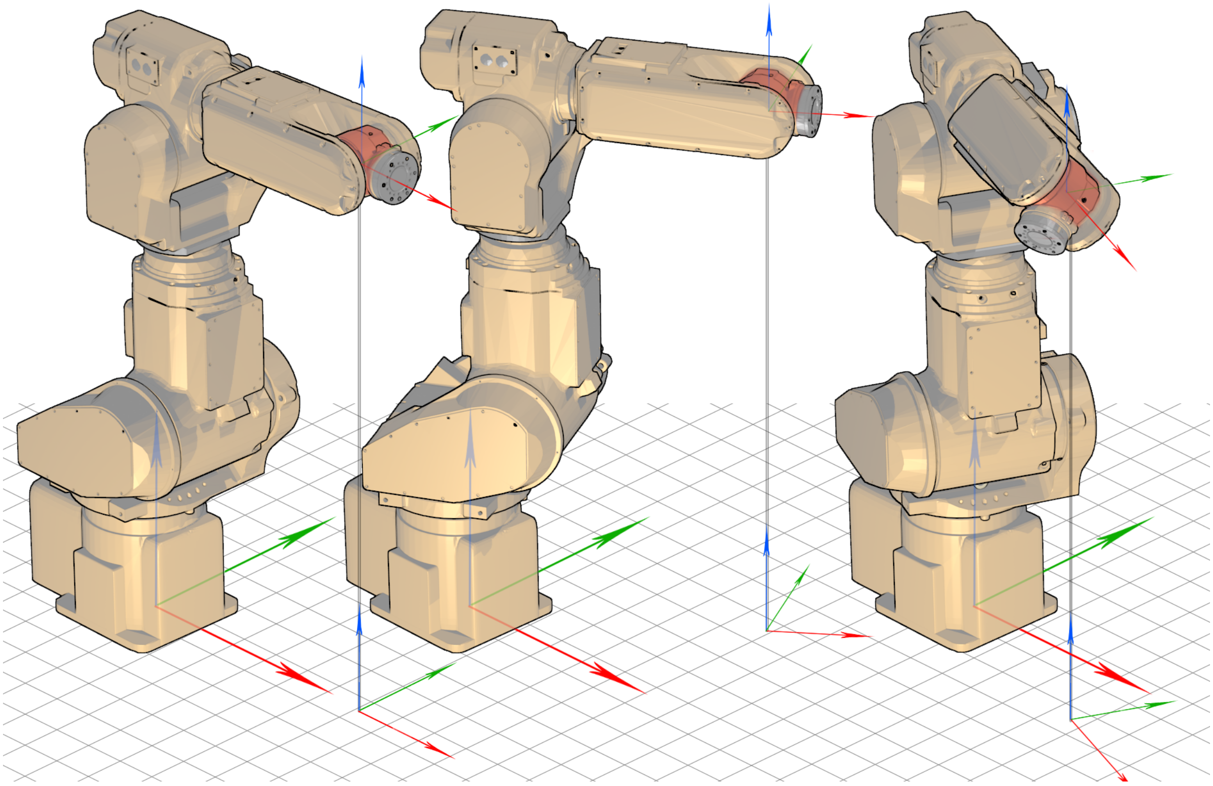


Figure 5.4: Illustration of the coupling coordinate system, and its relation to the position of the wrist.

As can be seen from the figure, the coupling is performed so that the change in the gripper analog's pose, from the time of activation, is related to a coordinate system that shares its z-axis with the robot base coordinate system. It is, however, rotated around z to give an x-axis that points radially out from the base towards the robot's wrist.

By using this coupling, the control is indiscriminate towards where in the robot's workspace the end-effector is placed when control is initiated. By seeing this in light of the placement of the remote base coordinate system depicted in figure 5.1, the result can be described as follows: Each time the remote operator activates control, the movements correspond to the perspective of the operator standing in the origin of the robot's base coordinate system and looking towards the robot's wrist. This is of course true for every time the control is activated.

5.3 Control Modes

In section 2.3, artificial fixtures and motion guidance was presented as an aid in unstructured environments and as a tool for increasing safety during remote operation.

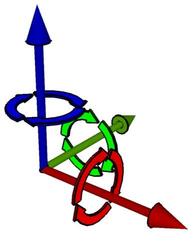
As the controller is held freely in space as previously described, the remote operator could struggle with controlling it perfectly according to his/hers intention. An example of this could be moving the controller, in a straight line. Therefore, in intricate tasks, it could be beneficial if the operator is able to mask out unwanted contributions from the control input. The approach used for achieving software based motion guidance, is to use a selection matrix, **S**, for masking out unwanted contributions to the control. The selection matrix is a 6x6 diagonal matrix where each element of the diagonal holds either 1 or 0. By multiplying the selection matrix with a 6-dimensional vector, the elements of resulting vector will be either kept or masked out depending on the value of the corresponding diagonal value of the selection matrix. Hence, by multiplying **S** with a 6-dimensional control input vector, $\Delta\mathbf{p} = [x, y, z, \omega_x, \omega_y, \omega_z]^T$, only the desired elements are kept, while the remaining elements are masked out.

Four control modes are mapped to one of the buttons on the external controller. For each button press, the next control mode is selected.

Three of the control modes are preset selection-matrices. The selection-matices associated with these are as follows:

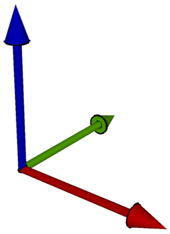
Full 6DOF, gives selection matrix 5.1:

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.1}$$



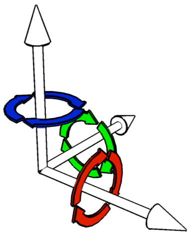
Translation only, gives selection matrix 5.2:

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.2}$$



Rotation only, gives selection matrix 5.3:

$$\mathbf{S} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$



The selection matrices are applied in tool coordinates, as will become apparent in the implementation chapter.

Reconfiguration
The last control mode, *reconfiguration*, utilizes the MR20's redundancy to perform self-motion, which is to keep the robot's tool stationary while changing its internal configuration. In this mode, the self-motion is proportional with the operator's movement of the gripper analog in the y-direction of the remote base coordinate system (see figure 5.1).

5.4 Operator Feedback Interfaces

This section presents the approach of the proposed solution for providing operator feedback. The focus is on a more conceptual level, the implementation is however used for outlining examples.

The tangible interfaces presented in section 5.1 aim to provide a physical link between the operator and the computer, and consequently the robot and its environment. The goal of the *operator feedback interfaces*, however, are to provide the operator with clear information pertaining to:

1. The remote robot's state and its environment.
2. The gripper analog and external controller's active control modes and parameters.

By providing this information, the assumption is that the operator can efficiently plan and execute relevant control strategies during remote operation. The three computer screens illustrated in figure 5.5, together with auditory feedback from speakers or headsets, aims to provide this information.

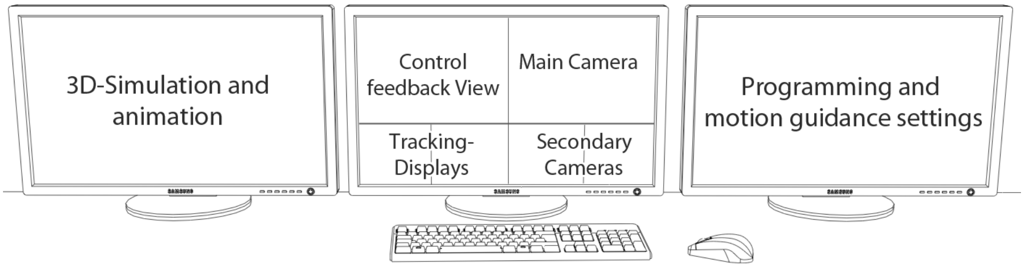


Figure 5.5: Illustration showing the arrangement of the displays available to the operator.

Before going into how the various displays are used, a high-level concept proposed for the operator feedback has to be explained. A layer, overlaying all displays[2], is transparent under normal control, but can be flashed or gradually color-tinted to convey information to the operator. The concept is intended for use in hazardous events such as an approaching collision, singularity, abnormally high contact-forces or personnel entering the robot cell. It can also be used for other events considered important, such as break-releasing etc. The intention of this, is to ensure that it is next to impossible to ignore information about hazardous events, which is a potential problem when using multiple displays and windows. The implemented example of this concept is a warning for an approaching singularity, and is further detailed later in this section as well as in 6.3.3.

5.4.1 Visual Feedback

Cameradisplay

The use of cameras during remote operation is nothing new, and has been used since the earliest days of telerobotics. In the proposed solution, three pan/tilt cameras are used. This allows the operator to aim the cameras to specific areas of interest in the local room.

As can be seen from the illustration in figure 5.5, camera displays are divided into one main- and two secondary camera-views. This is similar to the approach used in the CROP system presented in Thomessen et al. (2011). The premise for considering this a reasonable approach, is that some views may be more important than others, thus should take precedence over less important ones. It is also based on the assumption that the operator will experience orienting reflexes based on stimulus from his peripheral vision (see section 2.3) if an important event should occur on the secondary cameras.

Tracking display

The tracking display portion of the interface is intended to give the operator an understanding of the workspace available for operating the gripper analog, for example by displaying the field of view in vision-based tracking. A secondary function is verifying that the information pertaining to the tracking of the gripper analog is correct. This portion of the interface could differ or may be unnecessary, depending on the methods used for tracking.

---

[2] In concept - In the practical implementation, this has only been applied to the middle display.

The practical implementation uses a hybrid-system of a Microsoft Kinect and a MARG-sensor for obtaining the position and orientation, respectively. In light of this, the tracking display-portions of the interface include a depth-map of the Kinect-based tracking, where the tracked point is highlighted as a red dot. In addition, a 3D-model of the gripper is oriented based on the data supplied from the MARG-sensor. The displays allow the operator to see if the Kinect has acquired an appropriate tracking, and if the MARG-sensor is experiencing drift or interference, which, in section 3.2.2, have been pointed to as weaknesses when utilizing vision- and inertial-based tracking.

Control feedback view

The purpose of the control feedback view is to provide information about, and feedback from, the external controller. As presented in 5.1, each button on the external controller may have multiple functions and/or settings that the operator can cycle through, thus creating the need for showing these in an easy-to-understand way.

The approach used for this in the practical implementation, is to use icons for the functions and place these in a configuration that coincides with the placement of these on the physical controller.

In order to avoid confusion in the further part of the thesis, the following terms are defined:

Button Display - Refers to the implementation's software module for providing visual feedback of the external controller's state.

Button Interface - Refers to the combination of the external controller and the button display.

Figure 5.6 shows the button display in idle, meaning that the user is currently not cycling through any of the available modes, saving points or changing gripper-instructions.

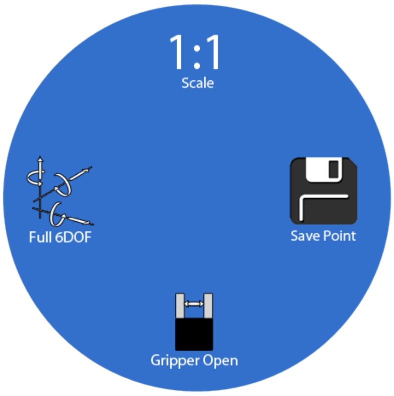


Figure 5.6: Figure of the button display, where the icons represent the active mode set by the external controller.

As the operator pushes any of the buttons on the external controller, the display expands to reveal the functions the button is cycling through for each button press. Figure 5.7 shows the fully expanded button display, that is, if the operator were to press all four buttons at the same time.

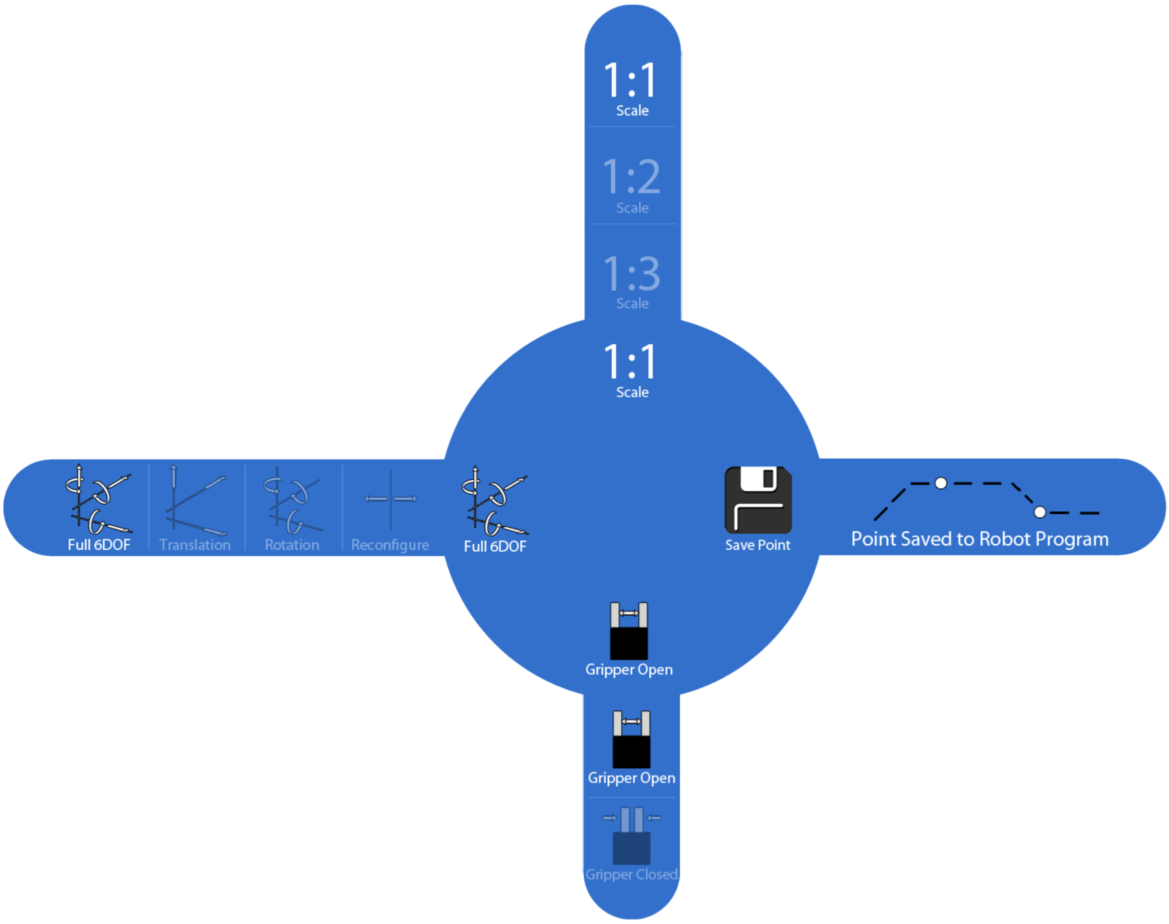


Figure 5.7: Figure of the button display fully expanded.

3D-Simulation and animation

The 3D-Simulation is to hold a precise 3D-model of the remote operated robot in its environment. The robot's joints are fully animated according to the movements of the physical robot. The simulation provides the operator with an alternative viewport for investigating the remote environment, while being free of concerns for camera-problems such as insufficient lighting, image-noise or slow frame-rates. It also offers the operator an arena for offline training and opportunity to get to know the control scheme without any repercussions. Lastly, if the 3D-models and kinematic models are precise enough, the operator could perform offline programming using only the 3D-Simulation view.

Programming and motion guidance settings

The function of this portion is two-fold:

1. Assist the operator in making robot-readable code.
2. Give the operator the manually set constraints for the control.

Each time the *save point* button on the external controller is pressed, the current position is added to a list-view for the operator to inspect.

The operator can also customize parameters pertaining to the saving of points such as:

- Interpolation type
- Smoothness
- Acceleration
- Speed

As previously presented, there are only three preset selection-matrices that can be chosen from the external controller. The motion guidance settings gives the operator the ability to manually override the selection-matrices instructed from the external controller. In addition, upper and lower limits for x,y,z,ω_x,ω_y and ω_z in both tool and base coordinates can be applied. These functions can be useful for restricting movements to one axis, plane, box etc.

Singularity warning using sensor-bridging
As has been presented earlier in this section, a concept of a visual overlay for conveying approaching hazards is proposed.

One type of singularity occurs when two (or more) of the robot's joints rotation axes are coaxial. When coaxial, the two joints will give the same contribution to the system. The robot manipulator is degenerated with one or more degrees of freedom in Cartesian space. Since the two joints are coaxial, the sum of their angles can be given by infinitively many combinations, leaving the system underdetermined. This might cause the servos to spin very fast, which could be damaging.

For the NACHI MR20, an example of singularity is as $\angle q_5 \to 0$, which makes q_4 and q_6 coaxial as shown in figure 5.8.

The implemented example of this concept uses the $\angle q_5$ and compares it to 0. If it is within a given threshold, the visual overlay is gradually tinted red, proportionally with how close the singularity is.

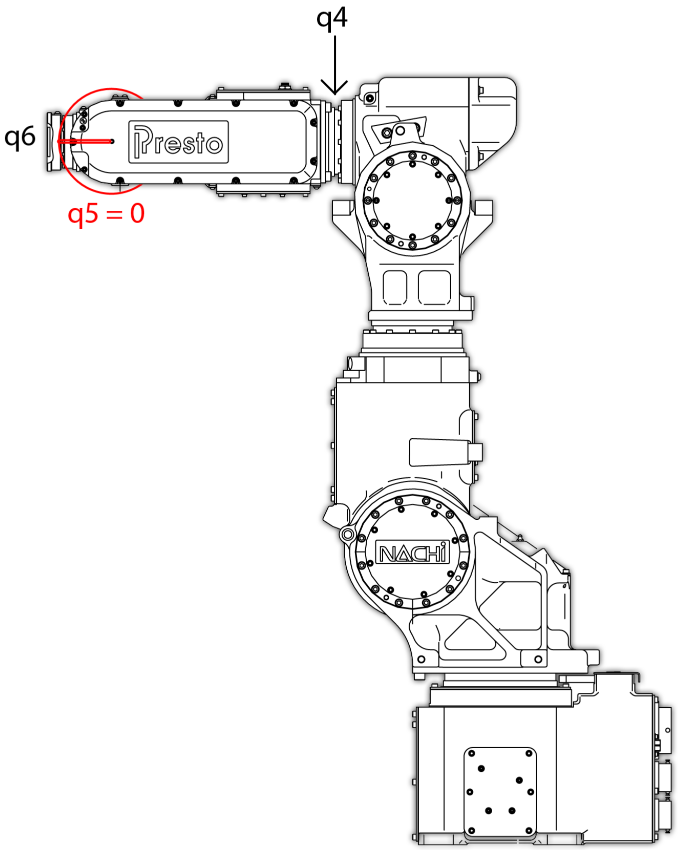


Figure 5.8: Example of singularity where the angle of $q_5 \rightarrow 0$ causes q_4 and q_6 to become coaxial.

5.4.2 Auditory Feedback

For an experienced operator, sounds such as the motor breaks releasing and servo-sounds etc., are what could be considered earcons. Experienced operators hear these sounds often, within the same contexts, and therefore immediately understand the message that the robot is active and/or moving.

The implemented examples of auditory feedback are sounds:

- Motor breaks on/off
- Gripper open/close

Chapter 6

Implementation of Cognitive Remote Operation System

In this chapter, details of the practical implementation of the Cognitive Remote Operation System will be presented.

The major part of this chapter is divided into the **Remote Room** and the **Local Room**. In these sections, implementations, both in terms of hardware and software for the respective locations, will be described.

6.1 Framework

In interest of full disclosure, this section will present various software components that has already been developed and/or implemented at PPM AS' laboratories and further expanded upon or used in any capacity by the candidate. A short description of the usage and important modifications of these components will also be presented.

LabView OLIMEX Framework
The LabView OLIMEX Framework was developed by PhD candidate, Balazs Daniel. It consists of connection handling of UDP connections to both an *ATI Industrial Automation Delta F/T Transducer* and the *Olimex real-time interface*, including initialization, read- and, in the case of the latter, also write capabilities. In addition, initialization, read and write for the *Schunk PG-70* gripper's CAN interface is included.

The portions of the framework for communicating with the *Schunk PG-70* gripper, as well as the sending of encoder delta values to the *Olimex real-time interface* are the main portions used during this thesis. In addition, the framework has been expanded by the candidate to handle additional connections between the remote- and local room, as well as expansions for the gripper control.

Self-Motion Algorithm
A fellow student working at PPM, Audun Sanderud, has been doing his master thesis *"Task programming of redundant industrial robot"*. As a part of his work, a self-motion algorithm was developed. This algorithm has also been adopted in the remote operation system. Self-motion in redundant robotics, is a motion that does not affect the robot's primary task space, consequently allowing a primary task to be uninterrupted while the robot's internal configuration can be changed.

The governing principle of his algorithm, is to calculate a desired joint motion by using the pseudo inverse of the jacobian up to the robot's elbow. This joint motion is subsequently projected into the robot's null space, establishing a joint vector $q_{\aleph}$. This joint vector can be added to the original joint vectors without affecting the primary task space of the robot.

CROP System components
The original Cognitive Remote Operation System (CROP), is a system developed at PPM AS' laboratories as a part of the HUNOROB (HUNOROB, 2012) project, and is presented in a paper by Thomessen et al. (2011). It consists of a camera-, animation-, text/graphic- and a sound module. The portion used in this thesis is the animation module. It is important to emphasize that in this thesis, the use of the term *Cognitive Remote Operation System* refers to the new system implemented by the candidate, unless *original* is emphasized.

In order to provide a clearer picture of what has been done to the original animation module, a short explanation of its original components are presented. It can be divided into three core components:

1. **VirCA - Virtual Collaboration arena**
   VirCA is according to its website (VIRCA-Website, 2012):

   > A loosely coupled modular, 3D Internet based Interactive virtual environment for collaborative manipulation of robots and other hardware or software equipment.

   In the case of this thesis, the VirCA environment is the front-end for 3D representation of the PPM AS laboratories with the Nachi MR20 robot.
2. **Nachi MR20 Cyberdevice RTC**
   VirCA uses Robotics Technology Middleware (RTM), which is a common platform standard for robots based on distributed object technology. One can therefore add objects into the VirCA environment by programming so-called Robotic Technology Components (RTC). The original CROP system therefore includes this RTC for adding the Nachi MR20 into VirCAs virtual environment, with a full robot model with all its rotational joints.
3. **Nachi Remote Server**
   The Nachi Remote Server component is, in the original CROP-system, the glue between the real robot and the Nachi MR20 Cyberdevice RTC. Consequently, it is the

component that controls what joint configurations are to be shown on the robot model in VirCA.

In the resulting Cognitive Remote Operation System proposed in this thesis, the three components presented above are used. However, there has been done considerable modifications to the Nachi Remote Server component for enabling the candidate to send joint-angles from LabView using UDP to the Nachi Remote Server for correct display in VirCA. This work was done by one of the systems original developers, MSc Laszlo Nagy, together with the candidate.

6.2 System Structure

This section aims to give a brief overview of the structure of the Cognitive Remote Operation System. Figure 6.1, shows a simplified overview of the various software- and hardware components of the system.

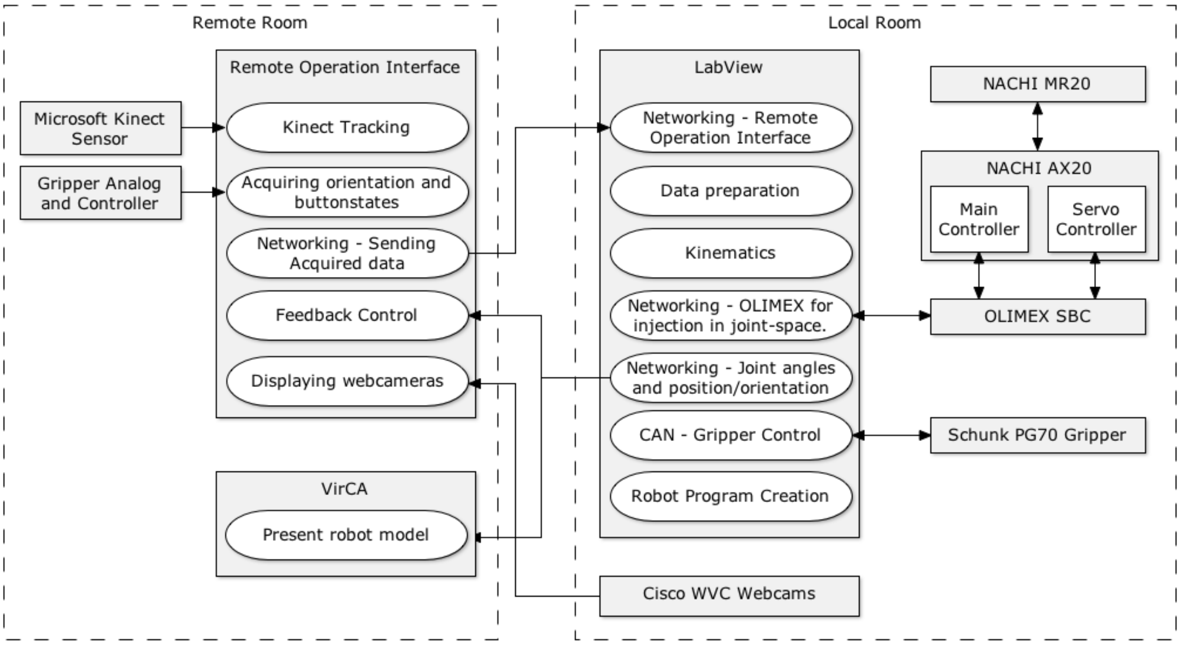


Figure 6.1: Simplified structure of the Remote Operation System

The system is distributed so that calculations pertaining to all manipulation of the robot and equipment are done at the local room. Calculations for acquiring information about the gripper analog's pose and control commands are done in remote room, as well as gathering and utilizing information about the robot and control for providing feedback to the operator.

6.3 Remote Room

Through the upcoming subsections, the hardware and software pertaining to the remote room will be explained in further detail. Figure 6.2 illustrates the setup of the cognitive remote operation system in the remote room.

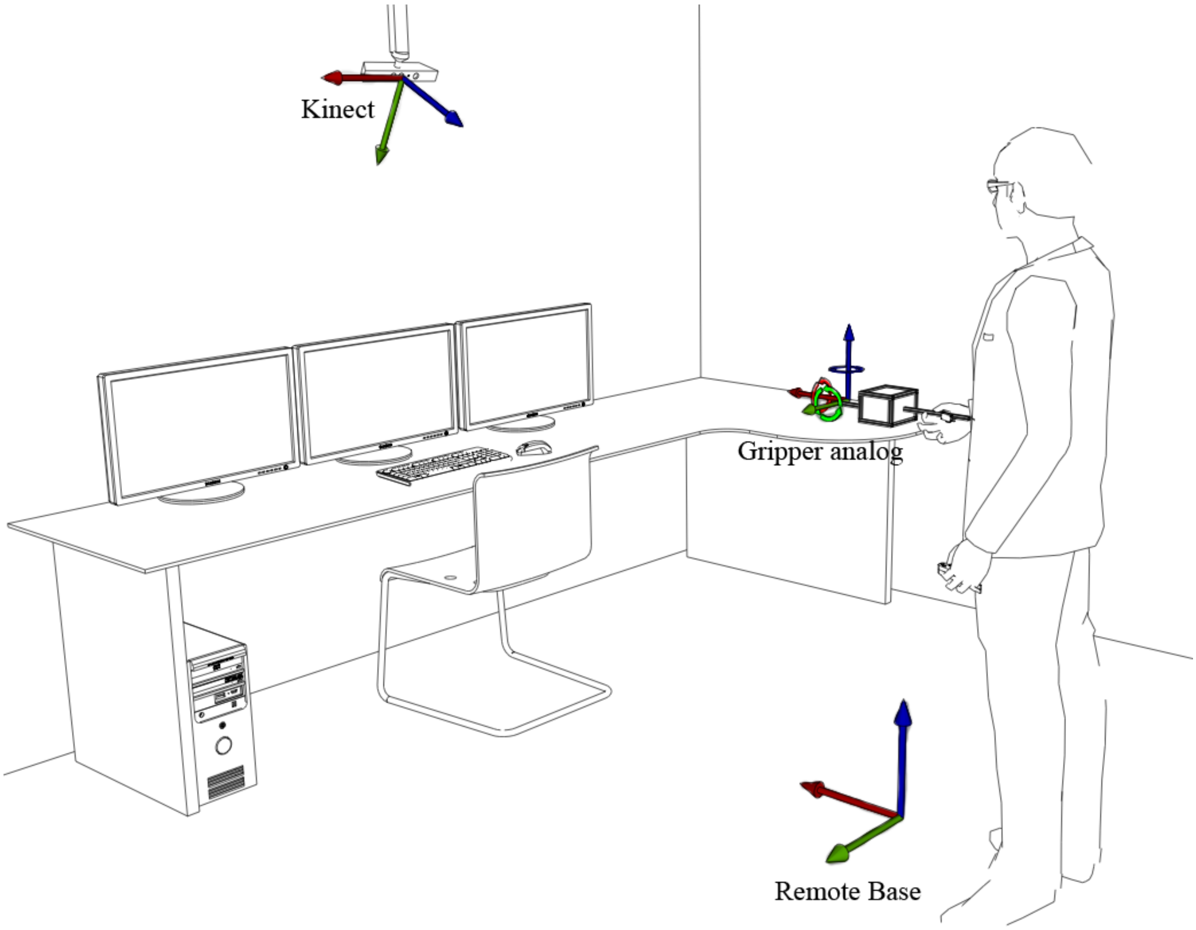


Figure 6.2: Illustration of the remote room.

For the remainder of the thesis, the term *remote base*, will refer to the coordinate system illustrated in figure 6.2.

6.3.1 Hardware

This section will present the hardware used at the remote room. An short introduction to these was done in section 1.3.

- Microsoft Kinect Sensor
- Arduino Uno microcontroller
- SparkFun 9DOF Sensor Stick
- Workstation Computer with three 27" screens

Gripper analog and external controller
In accordance with the proposed control scheme discussed in chapter 5, an analog to the end-effector in question, the Schunk PG70 gripper, was made. Its main structure consists of 1x1 cm cross-section T-slot beams which is assembled to loosely match the real grippers physical dimensions. Its walls are made of sheets of polycaprolactone plastic, which is a low melting-point, hand moldable plastic often used for prototyping purposes.

In addition to the gripper analog, an external four-buttoned controller was made. The four buttons was mounted on a prototyping board and its casing was made by polycaprolactone plastic. Depicted in figure 6.3, is the gripper analog and the external controller.

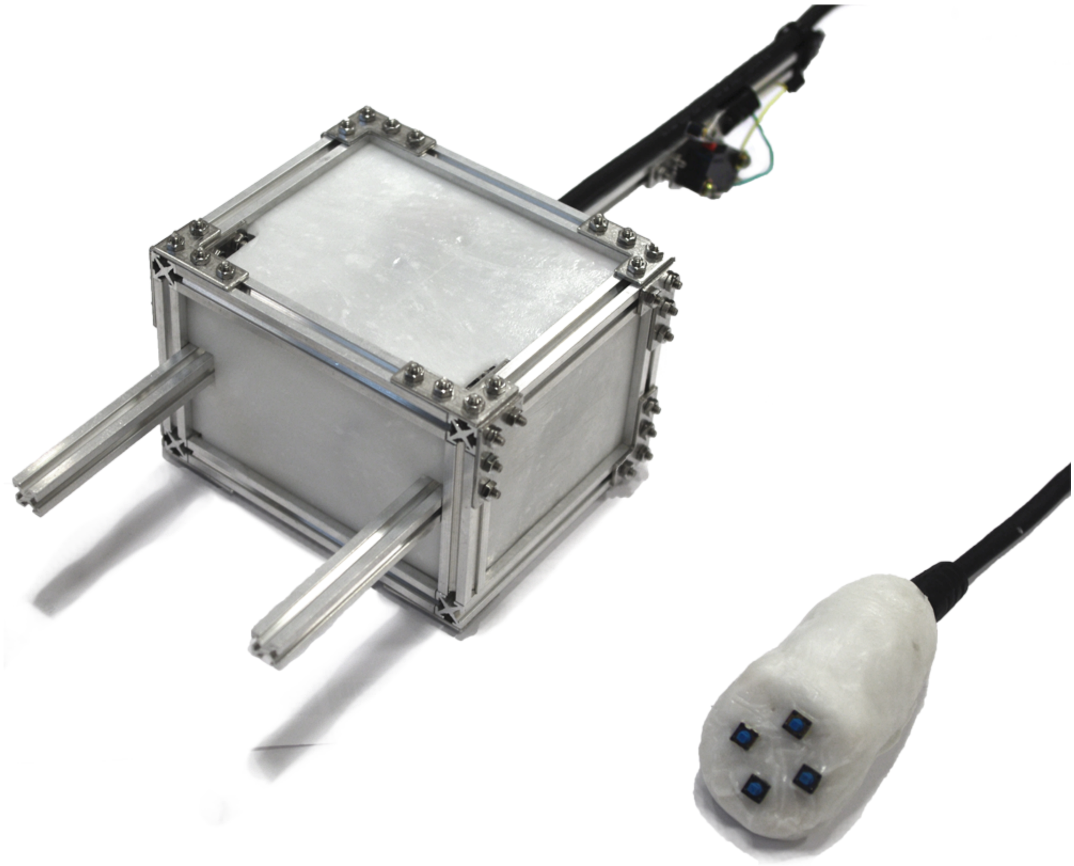


Figure 6.3: Gripper analog and external controller

Placed inside the gripper analog, is the Arduino Uno microcontroller, SparkFun 9DOF Sensor Stick and circuits for all the buttons. Illustrated in figure 6.4, is the electrical diagram for the circuits. In the remainder of the thesis, the term *MARG-sensor* (Magnetic, angular rate, gravity), will be used for the SparkFun 9DOF Sensor Stick.

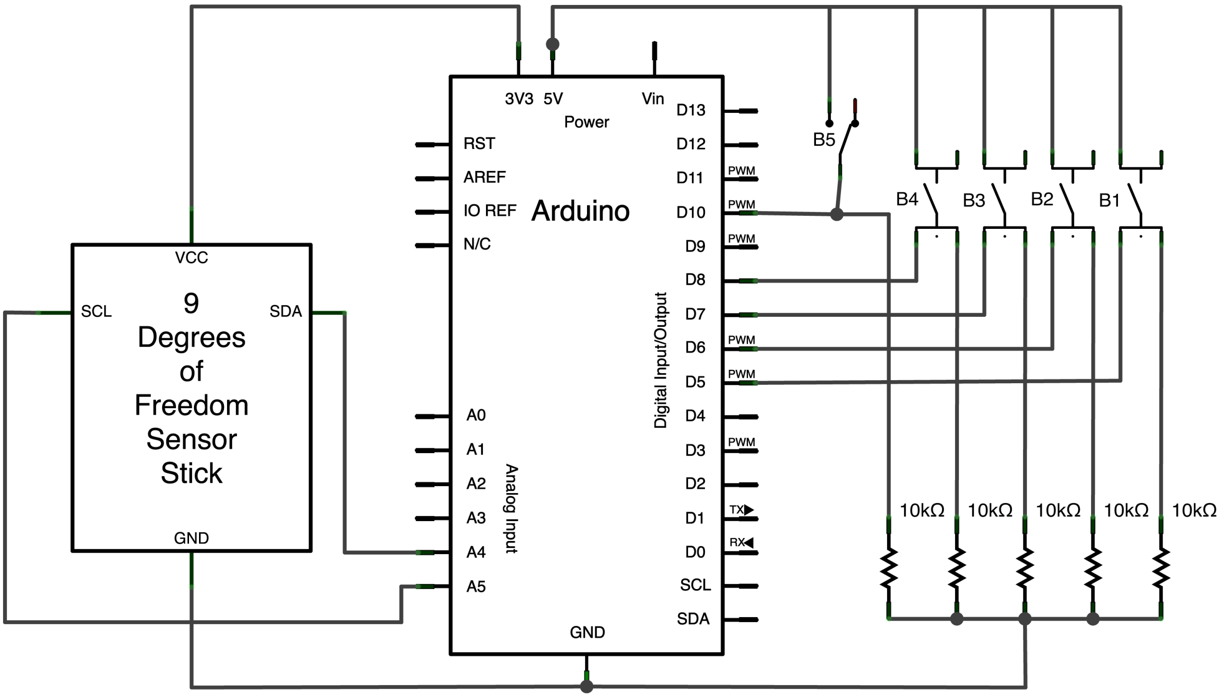


Figure 6.4: Electrical diagram for the gripper analog and external controller

Microsoft Kinect Sensor

Through a tracking algorithm, the Microsoft Kinect sensor is intended to provide the position of the gripper analog. By combining this, and information from the internal sensors in the gripper analog, its full pose can be determined. As previously mentioned, the Kinect projects a coded pattern of infrared dots into the space, and by analyzing the deformations in this pattern, a depth map is established. The fact that it uses infrared, makes the Kinect's depth map unaffected by changes in lighting conditions in the remote room.

The Kinect is mounted to the ceiling in the remote room and points down with an 28° angle. A normal operating area with this setup is at distances between 1-3m from the Kinect. In an article by Khoshelham and Elberink (2012), empirical tests showed that:

- The Kinect's random error of depth measurements increased quadratically with the distance from the sensor, from a few millimeters at 0.5m to 4cm at 5m.
- The depth resolution decreases quadratically with the distance, where point spacing increased from ≈ 2mm at 1m to ≈ 7cm at 5m.

It was found that the theoretical random error was given by:

$$\sigma_Z = \frac{2.85 \cdot 10^{-5} cm}{2} \cdot Z^2$$

and depth resolution by:

$$\delta_Z = 2.85 \cdot 10^{-5} cm \cdot Z^2$$

where Z is the distance from the sensor along its optical axis.

This gives a theoretical random error between $\approx 0.14cm - 1.28cm$ and a depth resolution between $\approx .028cm - 2.6cm$ for the setup used for the remote operation system.

6.3.2 Microcontroller Software

The Arduino Uno is running a modified version of a firmware called Razor AHRS (Appendix F for details), which is a firmware that uses the MARG-sensor to become a Attitude and Heading Reference System (AHRS), thus providing absolute orientation of the MARG-sensor in space.

The algorithm performs complimentary filtering between the accelerometers, gyros and magnetometers similarly to the simplified explanation described in section 3.2.2. For the firmware to function properly, sensors have to be calibrated to establish the baseline gravity readings for all accelerometers, stationary drift in the gyros and the baseline magnetic readings for all axes. The resulting calibration values can be found in appendix G. The AHRS-algorithm updates the orientation at $\approx 50Hz$.

In addition to the Razor AHRS firmware, the Arduino was programmed to perform readings of all the buttons on the external controller. It was programmed to give the following button state behavior: (ref. electrical diagram in figure 6.4)

B1 - Is the *control mode selection* button (left on the controller). It has four states, giving the following behavior: $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$....

B2 - Is the *gripper command* (down on the controller). It has two states, giving the following behavior: $0 \rightarrow 1 \rightarrow 0$....

B3 - Is the *save point* button (right on the controller). Programmed to be momentary: $\rightarrow 1$ when pressed, otherwise 0.

B4 - Is the *scaling preset* button (up on the controller). It has three states, giving the following behavior: $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$....

B5 - Is the main switch located on the gripper analog. Programmed to be momentary $\rightarrow 1$ when pressed, otherwise 0.

The results from the Razor AHRS-algorithm and button states are concatenated into a string with the following structure:

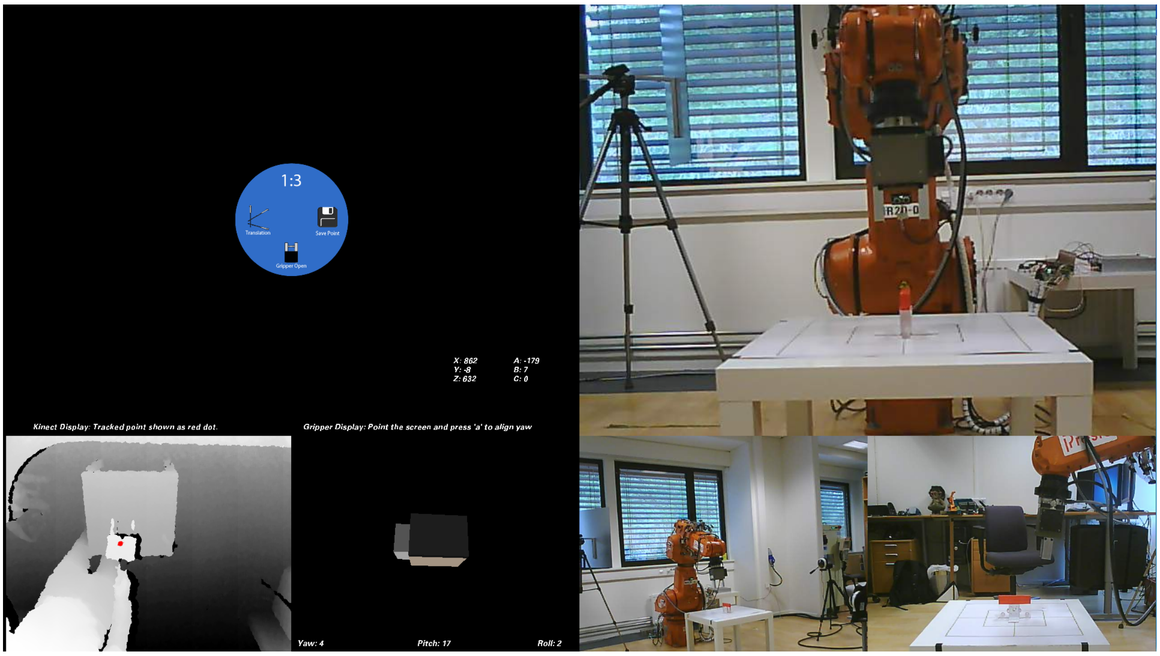
"ω_z, ω_y, ω_x, `B1, B2, B3, B4, B5`"

Where ω_z, ω_y and ω_x are in degrees and B1-5 are integers as previously described. The resulting string is then sent via a serial connection to the *Remote Operator Interface* at a baud rate of 57600.

6.3.3 Remote Operator Interface

The *Remote Operator Interface* (ROI), is an application made using Processing[1]. At the front-end, it delivers the functionality of the middle display in figure 5.5 in section 5.4, such as control feedback, tracking displays and camera displays. In addition to this, it is also responsible for the Kinect tracking itself, as well as the communicating with the gripper analog. Furthermore, it is responsible for structuring and relaying this information to the local room.

In presenting the ROI, its overarching structure will be used as the starting point, where mainly parts pertaining to acquisition and relaying information will be described in detail. Figure 6.5 is a screenshot from the Remote Operator Interface.



X: 862
Y: -8
Z: 632

A: -179
B: 7
C: 0

Kinect Display: Tracked point shown as red dot.

Gripper Display: Point the screen and press 'a' to align yaw

Yaw: 4

Pitch: 17

Roll: 2

Figure 6.5: Screenshot of the Remote Operator Interface.

---

[1] An open source programming language and environment based on the Java language.

The ROI can be divided into the following modules:

Data Handling - The portion that is responsible for obtaining the information from the gripper analog via a serial connection, as well as the robot's joint-angles and pose of its end-effector. It is also responsible for relaying all information to the local room.

Feedback Control - It controls the auditory feedback and the visual overlay feedback presented in section 5.4.

Kinect Tracking - It runs the tracking algorithm as well as displays the active tracking point on its depth-map.

Gripper Display - It displays a 3D-model of the gripper analog for the user to verify that the offset in the grippers yaw, ω_{zo}, is correct (the offset, ω_{zo}, will be further explained on the next page.).

Button Display - The portion of the application responsible for displaying and animating the button display described in section 5.4.1 and depicted in figure 5.6 and 5.7. This also holds a small numeric display for the pose of the robot's end-effector.

Webcamera Display - This is responsible for displaying three MJPEG web camera streams.

Some of these modules will not be further elaborated upon, as the many of the aspects of the Remote Operator Interface of interest for this thesis have been discussed section 5.4. However, the Kinect Tracking, Data Handling and Feedback Control will be explained in greater detail.

Kinect Tracking
The position tracking of the gripper analog utilizing the Kinect-sensor, is for this thesis to be considered a black box. However, in interest of full disclosure, some background into what has been used to for the tracking will be briefly presented in this section.

The tracking is done by running an example provided in the SimpleOpenNI library (see Appendix F for more details). The example is made for tracking the tip of the user's hand, and the tracking is initiated by a wave-gesture. When waving the gripper analog in front of the Kinect, it gets a fairly robust lock on the analog. However, the position of the tracking point is not completely as desired, and has to be compensated for. A further description of this, as well as an approach for compensation which will be elaborated on in section 6.4.7.

Data Handling

There are mainly two input streams of data to the Remote Operator Interface.

Gripper analog and controller-input
This is the text string presented in section 6.3.2. Internally in the ROI, this is used for feeding the gripper- and button display. It is also used as the basis for one of the output strings.

LabView Server-input
This is a text string received from the LabView Server with the robot's joint-angles and its end-effector's pose, that is structured in the following manner:

"q_1, q_2, q_3, q_4, q_5, q_6, q_7, x, y, z, ω_x, ω_y, ω_z"

This information is used to display the robot's end-effector pose in the ROI, and q_5 is used for the visual feedback of an approaching singularity, which will be further explained in later in this section.

Now, moving over to the output-side, there are two streams:

LabView Server (Gripper analog)-output
In addition to the contents of the string received over serial from the gripper analog, the analogs offset in yaw, ω_{zo}, is introduced to it. This offset, is the offset in the grippers yaw that aligns the MARG-sensors coordinate system to the remote base coordinate system (see figure 6.2). This is also used in the preparation of the orientation data in section 6.4.6. The resulting output string sent to the LabView Server is:

"ω_z, ω_y, ω_x, `B1, B2, B3, B4, B5,` ω_{zo}"

LabView Server (Kinect Tracking)-output
The refresh rate of the Razor AHRS-algorithm is, as mentioned $\approx 50Hz$. The Kinect Tracking is however running at $\approx 30Hz$ (refresh rate of the Kinect). Therefore, the position obtained from the Kinect tracking is put in a separate output thread to ensure that the information from the gripper analog is not throttled in any way. The string of this output is:

`"x, y, z"`

Feedback Control

As previously mentioned in section 5.4.1 & 5.4.2, there is auditory feedback and a visual overlay implemented for providing the operator feedback of various actions. The auditory feedback includes sounds for the motor breaks turning on and off, as well as sounds for gripping. The implementation of this could be considered relatively simple in this thesis, as there is no frequency modulation or other manipulation of the audio being performed. This will therefore not further elaborated on in this section. Instead, the approach for providing the operator indications for an approaching singularity will further elaborated on. A screenshot showing the behavior of overlay the overlay is depicted in figure 6.6.

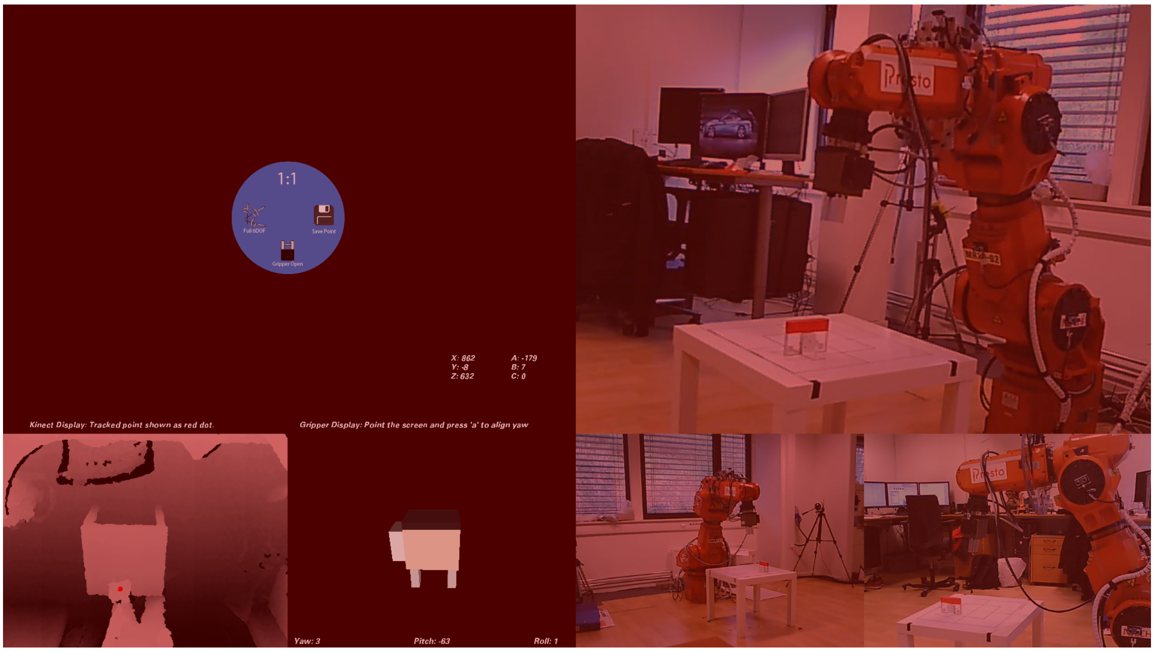


Figure 6.6: Behavior of nearing singularity simulated in the Remote Operator Interface.

The warning overlay is a tint-layer that stretches over the whole Remote Operator Interface. The usage and parameters of the tint layer in the Processing programming language is:

```
tint(value1, value2, value3, alpha)
```

where value1-3 can be used as either RGB[2] or hue/saturation/brightness and alpha is the opacity.

In the implementation, the RGB + alpha is used. The algorithm compares the current value of q_5 against a threshold for the warning to start, q_w, which has been set to 28°. The algorithm governing the tinting is presented in figure 6.7

---

[2] Red,Green,Blue

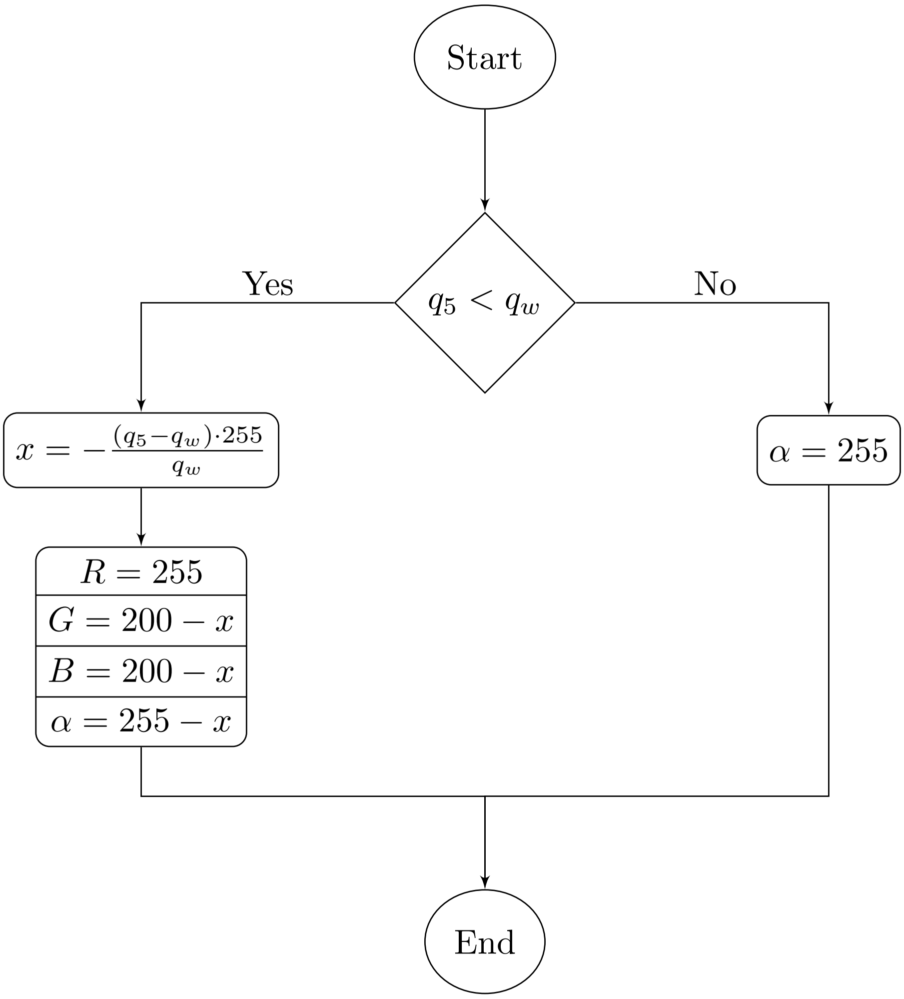


Figure 6.7: Block diagram for the tinting of the overlay according to an approaching singularity.

As can be seen from the algorithm for the tint-values in figure 6.7, the screen is tinted proportionally with how close the q_5 is to zero. When the joint angle is above the threshold, the opacity of the layer is set to maximum. When the joint is within the threshold however, red (R) is set to maximum, while green (G) and blue (B) are taken from 200 → 0. This gives a more subdued red (closer to white) in initial stages of the tinting.

6.4 Local Room

The local room is the location of the physical robot system. All calculations regarding the manipulation of the robot and equipment are performed at this location. The following subsections will describe, among others, hardware, kinematics, preparation of data, control structure and control algorithm.

6.4.1 Hardware

Of the hardware presented in section 1.3, the following are for use at the local room:

- NACHI MR20 7 axis industrial robot
- NACHI AX20 controller
- Olimex interface
- Server computer
- Three Cisco WVC210 Pan/tilt zoom cameras.

The Olimex interface could be considered the glue between the server computer and the NACHI AX20. It is capable of manipulating the data going between the NACHI AX20's servo- and main controller board, consequently enabling control of the robot based on third-party code. Figure 6.8 shows the principle function of using the Olimex interface for this purpose.

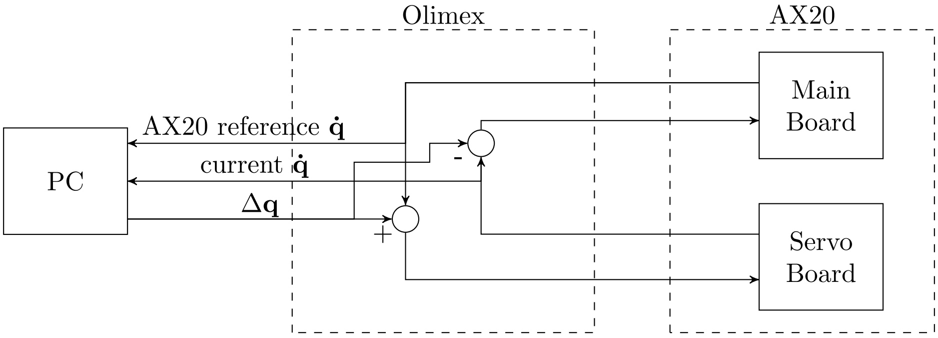


Figure 6.8: Principle function of the Olimex Interface.

6.4.2 Kinematics for NACHI MR20

The NACHI MR20 (hereby referred to as MR20) is a seven-axis robot. This gives new opportunities but also new challenges compared with standard six-axis robots. With seven axis there will be infinitely many solutions for most positions and orientations. The approach used for handling the MR20's seven axis, is to simply ignore one of them during position control, while using a kinematic model for all seven axis for performing self-motion. This section will present the robot definitions used for this in the implementation for the MR20. As the Denavit-Hartenberg convention has been used for all link definitions, a short presentation of this convention will follow.

Denavit-Hartenberg

The Denavit-Hartenberg (DH) formulation makes it possible to develop the transformation matrix based on four parameters per link, compared to the six needed in homogeneous transforms. In DH formulation one defines the links between joints, rather than the joints themselves. The rotation of the joint is always around the z-axis, and the x-axis is always normal on the current and last z-axis. see Figure 6.9

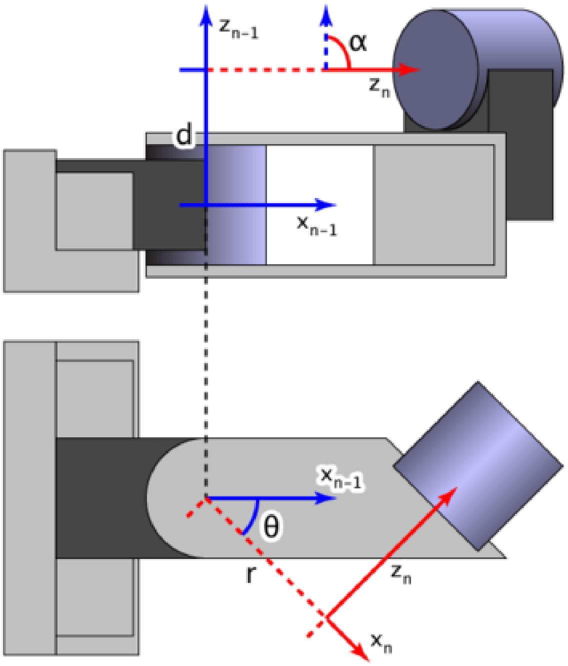


Figure 6.9: The Denavit-Hartenberg Formulation (Wikimedia Commons 2012)

Where the link parameters are defined as follows:

Rotation	θ_n	rotation along Z_n of X_{n-1} into X_n
Offset Distance	d_n	displacement along Z_n from X_{n-1} to X_n
Length	r_n	displacement along X_n of X_{n-1} from Z_{n-1} to Z_n
Twist Angle	α_n	rotation about X_{n-1} of Z_{n-1} to Z_n

Using the standard homogeneous rotation matrices for rotation about x-,y- and z-axis (see appendix B), the corresponding transformation matrices for the link parameters can be developed as follows:

$$\mathbf{T}_{d_n} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.1}$$

$$\mathbf{T}_{\theta_n} = \begin{bmatrix} cos(\theta_n) & -sin(\theta_n) & 0 & 0 \\ sin(\theta_n) & cos(\theta_n) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.2}$$

$$\mathbf{T}_{r_n} = \begin{bmatrix} 1 & 0 & 0 & r_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.3}$$

$$\mathbf{T}_{\alpha_n} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\alpha_n) & -sin(\alpha_n) & 0 \\ 0 & sin(\alpha_n) & cos(\alpha_n) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.4}$$

The product of these four basic homogenous transformation matrices yields the composite transformation matrix $\mathbf{T}_n^{n-1}$, known as the D-H transformation matrix for adjacent coordinate frames (Gonzalez et al., 1987).

$$\mathbf{T}_n^{n-1} = \mathbf{T}_{d_n}\mathbf{T}_{\theta_n}\mathbf{T}_{r_n}\mathbf{T}_{\alpha_n} \tag{6.5}$$

$$\mathbf{T}_n^{n-1} = \begin{bmatrix} cos(\theta_n) & -cos(\alpha_n)sin(\theta_n) & sin(\alpha_n)sin(\theta_n) & r_n cos(\theta_n) \\ sin(\theta_n) & cos(\alpha_n)cos(\theta_n) & -sin(\alpha_n)cos(\theta_n) & r_n sin(\theta_n) \\ 0 & sin(\alpha_n) & cos(\alpha_n) & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.6}$$

The total transformation matrix for the robotic manipulator with n links may the be formulated as:

$$\mathbf{T}_n^0 = \mathbf{T}_1^0\mathbf{T}_2^1\mathbf{T}_3^2 \ldots \mathbf{T}_n^{n-1} \tag{6.7}$$

Figure 6.10 shows a wireframe model of the MR20's seven links with its DH-parameters. In the remainder of this section, the two robot definitions used in the implementation, will be presented. Values used in the DH-matrices are based on the datasheet for the MR20 (Appendix H).

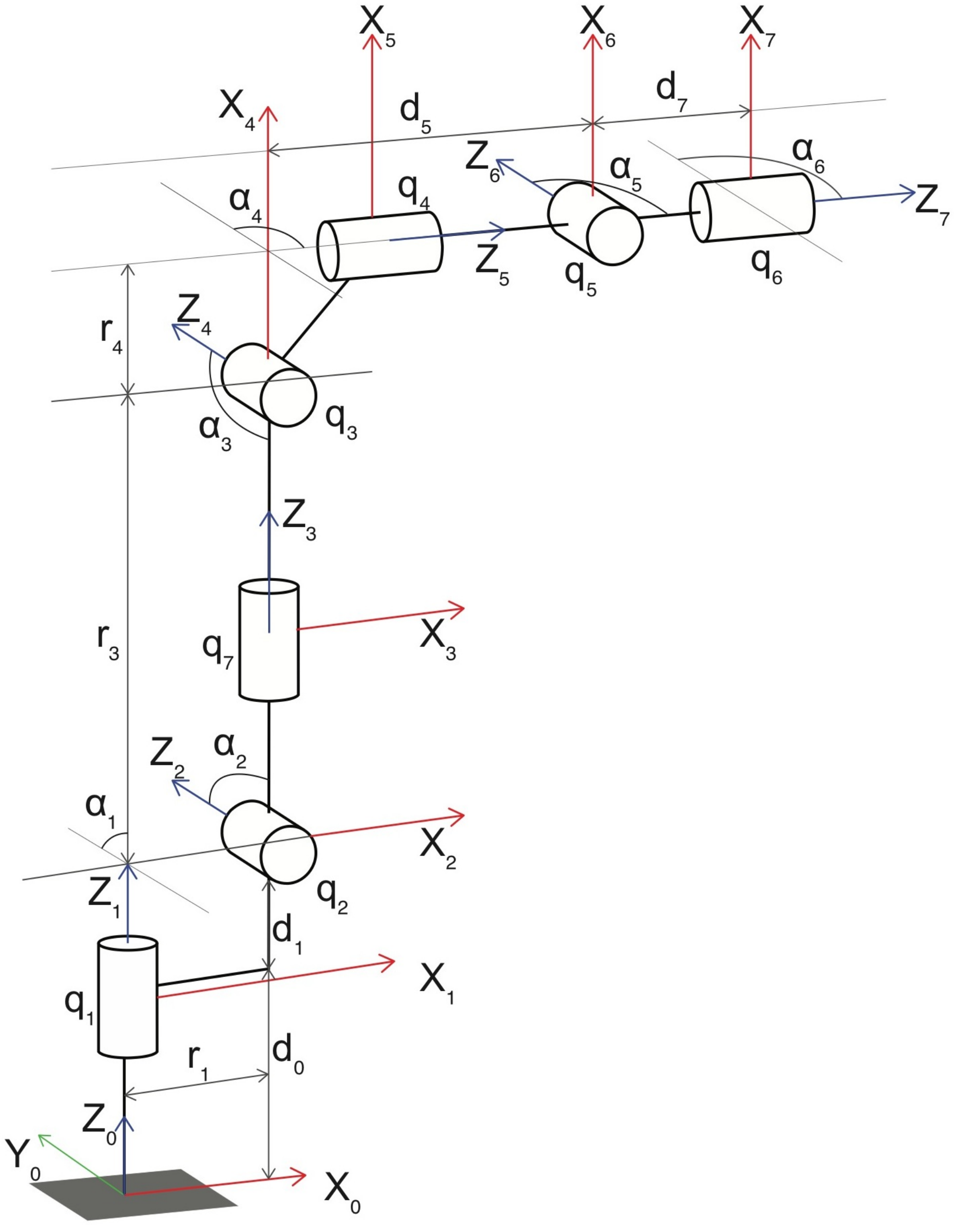


Figure 6.10: Wireframe model of the NACHI MR20's links.

As has already been touched upon in section 6.2, all calculations pertaining to the kinematics is done at the local room using *LabVIEW 2011*. For much of the kinematic calculations, the *LabVIEW 2011 Robotics Module* is used, where the portions belonging to forward- and inverse kinematics are based on Prof. Peter Corke's robotics toolkit for MATLAB (Petercorke.com, 2012).

Full seven-axis definition

In order to determine the MR20's null space for the self-motion algorithm, the jacobian of the full seven-axis has to be established. The definition used for this is presented in table 6.1.

Link	θ	d	r	α
Base	0	0,3	0	0
1	θ_1	0,2	0,15	$-\frac{\pi}{2}$
2	θ_2	0	0	$\frac{\pi}{2}$
3	θ_7	0,6	0	$-\frac{\pi}{2}$
4	$\theta_3 - \frac{\pi}{2}$	0	0,1	$-\frac{\pi}{2}$
5	θ_4	0,5	0	$\frac{\pi}{2}$
6	θ_5	0	0	$-\frac{\pi}{2}$
7	θ_6	0,175	0	0

Table 6.1: Denavit-Hartenberg Matrix for the MR20 with all joints

6-axis definition with q_7 as redundant

As discussed in the introduction to this section, a 6-axis definition of the MR20 is used for the position control. The kinematic definition uses q_7 as an inert redundant joint. This allows for the inverse kinematics to provide solutions based on $q_1 \to q_6$'s solution space, while q_7 is not ignored.

q_7 is chosen as the redundant joint, as its rotational axis is close to the one of q_1. Due to the distance $r_1 = 0.15$ (table 6.2, length of link 1), choosing $q1$ would reduce the robot's reach by this length. The definition used for this is presented in table 6.2.

Link	θ	d	r	α
Base	0	0,3	0	0
1	θ_1	0,2	0,15	$-\frac{\pi}{2}$
2	$\theta_2 - \frac{\pi}{2}$	0	0,6	θ_7
3	θ_3	0	0,1	$-\frac{\pi}{2}$
4	θ_4	0,5	0	$\frac{\pi}{2}$
5	θ_5	0	0	$-\frac{\pi}{2}$
6	θ_6	0,175	0	0

Table 6.2: Denavit-Hartenberg Matrix for NACHI MR20 where q_7, with a joint-angle of θ_7, is used as a redundant joint.

As can be seen from table 6.2, the rotation of the redundant joint, q_7, is inserted as a twist-angle, α, of link 2.

6.4.3 Servo Encoder Values to Radians

The data received from the Olimex, are servo encoder values. This creates the need for establishing the relationship between servo's encoder values and joint angles. This was done by programming the NACHI AX20 controller to set all the robot's joints to its "home position", which is illustrated in figure 6.11. *The remaining part of section 6.4.3 is an excerpt from previous work by the candidate and a fellow student in a specialization project (Reme and Sanderud, 2011).*

Figure 6.11: Home position.

Readings of the encoder values was performed in both the home position and a position where all joints were programmed to move an additional +10° in all joints. The results can be seen in table 6.3.

Joint	Hex	Long value	Angle
q1	80000	8388608	0°
q2	80000	8388608	90°
q3	80000	8388608	0°
q4	80000	8388608	0°
q5	80000	8388608	0°
q6	80000	8388608	0°
q7	80000	8388608	0°

Joint	Hex	Long value	Angle
q1	7FDA64	8378980	10°
q2	802600	8398336	100°
q3	802476	8397942	10°
q4	7FE500	8381696	10°
q5	7FE9C7	8382919	10°
q6	7FE955	8382805	10°
q7	7FDA64	8378980	10°

Table 6.3: Encoder values in home position to the left and encoder values for home+10° to the right.

Table 6.4 shows the scaling factors for each joint, found using the delta of the long values and corresponding angles for each joint, from table 6.3

Joint	Scalingfactor
q1	$-1.81276 \cdot 10^{-5}$
q2	$-1.76413 \cdot 10^{-5}$
q3	$-1.86986 \cdot 10^{-5}$
q4	$-2.52507 \cdot 10^{-5}$
q5	$3.06790 \cdot 10^{-5}$
q6	$-3.00763 \cdot 10^{-5}$
q7	$-1.81276 \cdot 10^{-5}$

Table 6.4: Scaling factors for the joints with rad/encoderlong as unit.

As the direction of rotation for some of the joints from the robot system did not coincide with the directions in the kinematic model developed in LabVIEW, the signs for some of the scaling factors had to be altered.

6.4.4 Control Structure

In this subsection, the principle structure of the robot's control is presented.

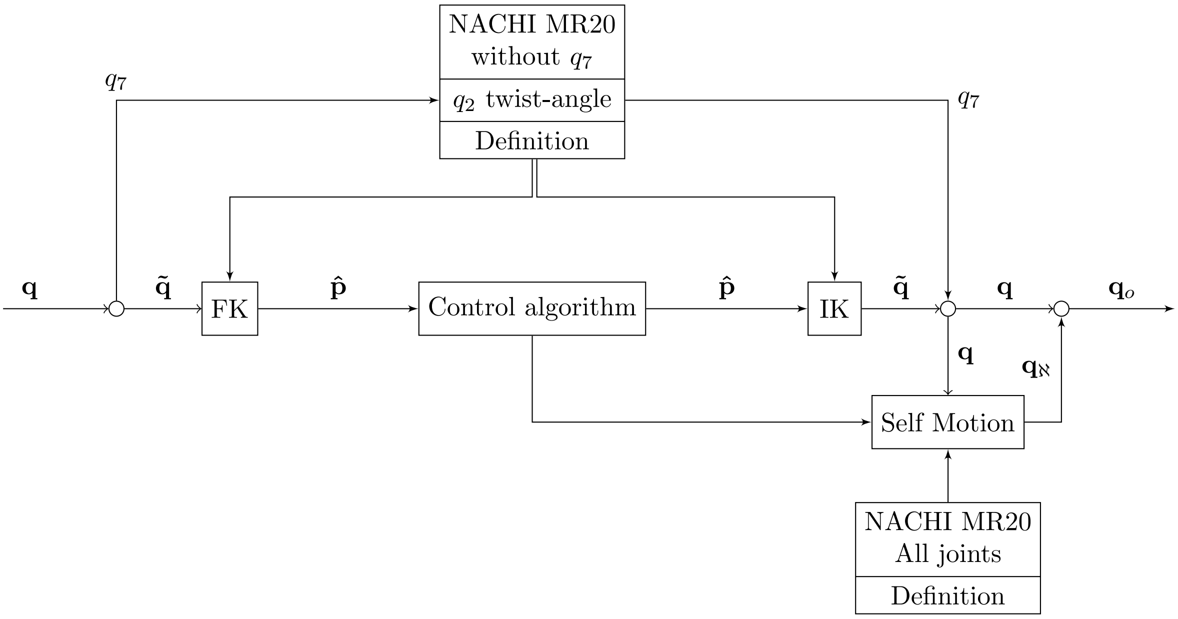


Figure 6.12: Block diagram of the control system for the industrial robot system.

Figure 6.12 shows how the control algorithm governing the manipulation of the robot is implemented. It allows for traditional 6DOF control using the 6-axis definition, presented in section 6.4.2, as well as self-motion by utilizing the algorithm presented in 6.1 and the full 7-axis kinematic definition.

The $\mathbf{q}$-input is the current 7-dimensional joint vector after the conversion presented in section 6.4.3. By the same token, the output, $\mathbf{q}_o$, has to be converted back to servo encoder values before it can be sent to the Olimex.

A prerequisite for this robot control system to function properly, is that the NACHI AX20's main controller board does not instruct the robot to move during the remote operation.

6.4.5 Control Algorithm

This section will present the implemented algorithm governing the manipulation of the robot during remote operation. In presenting the algorithm, the premise is that both the position- and orientation data from the gripper analog have been properly aligned to the coupling-point (the approach for this will be presented in section 6.4.6 and 6.4.7).

The control algorithm is only active if the main switch on the gripper analog is pressed down, signifying execution of one of the two main control-modes, which is specified by the operator:

- 6-axis position control where joint q_7 is inert.
- Self-motion

The main principle behind the position control, is to utilize the robot's 6 dimensional position-orientation vector, $\hat{\mathbf{p}}$, at the time of switch activation, t_s, as a baseline and add the change in the gripper analog's pose over the same timespan. If the control algorithm is not active, $\hat{\mathbf{p}}$ will simply be fed through.

In the interest of avoiding any inaccuracies or ambiguities, it is emphasized that when speaking of *delta* in the further part of this thesis, it is always in relation to the the time of activation, t_s.

Figure 6.13 shows an overview block diagram of the algorithm, which will be explained in further detail throughout this and following subsections.

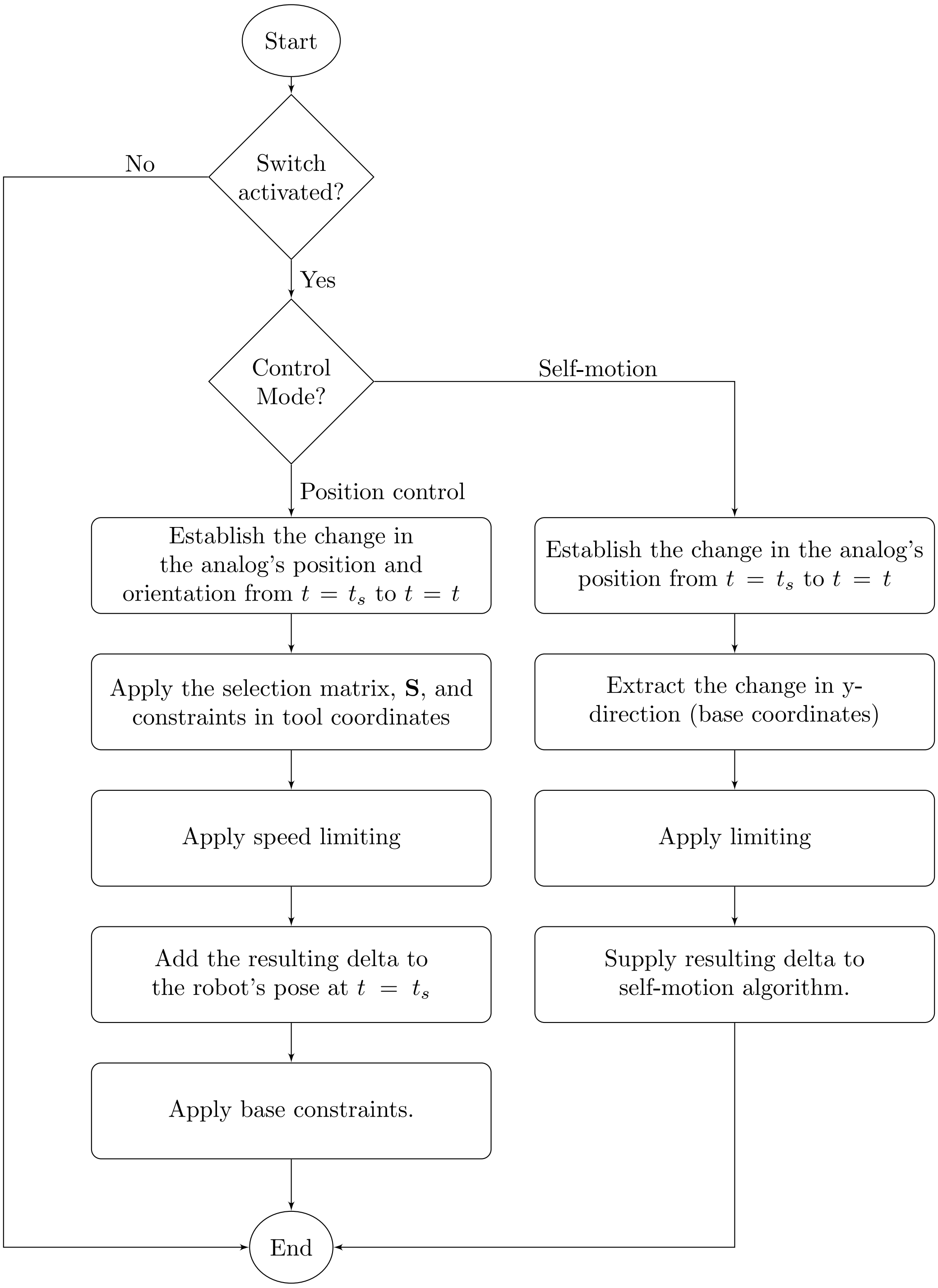


Figure 6.13: Block diagram of the control algorithm.

Position Control

When in the mode of position control, the change in the analogs pose is the first to be established:

$$\Delta\mathbf{r} = \mathbf{r}_{t=t_s} - \mathbf{r} \tag{6.8}$$

$$\Delta\mathbf{R} = \mathbf{R}\mathbf{R}_{t=t_s}^{-1} \tag{6.9}$$

Where $\mathbf{r}$ is the 3-dimensional position vector for the gripper analog's tool center point, $\mathbf{R}$ is the 3x3 rotation matrix for the gripper analog. The index of $t = t_s$ denotes that it is the value of the vector/rotation matrix at the time of the switch, t_s.

The resulting delta values are then transformed to the tool reference frame at $t = t_s$, where it is converted to a six-dimensional vector $\mathbf{p} = [x, y, z, \omega_x, \omega_y, \omega_z]^T$ and multiplied with the active selection matrix, $\mathbf{S}$, for masking out unwanted contributions to the control. In addition to the selection matrix, an additional algorithm for setting upper- and lower bounds for the contributions is applied in the tool reference frame. The limiting algorithm functions by the principle displayed in figure 6.14, where x_i denotes the i'th element of $\mathbf{p}$, while LB_i and UB_i is the respective elements' lower and upper limits.

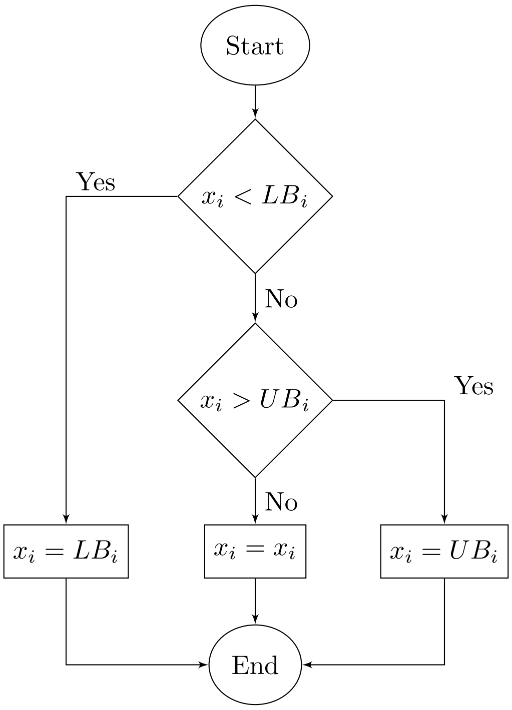


Figure 6.14: Block diagram of the limiting algorithm.

At this point, the results are transformed back to the base reference frame, and subsequently fed into a speed limiter, which will be presented in section 6.4.8. Lastly, the results are added to the robot's position at the time of the switch, $\hat{\mathbf{p}}_{t=t_s}$, where additional constraints are applied by the same algorithm as in figure 6.14.

The limits in the base coordinates can be useful for defining areas that are available for operation during remote operation of the robot, and is used in most experiments in chapter 8, for reducing the probability of collision with certain objects.

Self-motion

As can be seen from figure 6.13, the self-motion part of the algorithm has similarities with the position control mode. In short, only the translation of the gripper analog in y-direction relative to the remote base is of interest. It is subsequently fed into a limiter similar to the speed limiter that will be presented in section 6.4.8. However, the end result is not limited to any conventional measurement of speed, and is used more as a safety precaution and a practical solution for handling any unexpected failures in the tracking of the gripper analog.

Concluding the self-motion control mode, is to supply the self-motion algorithm with the limited value, which creates an appropriate joint vector $\mathbf{q}_\aleph$ in the robot's null space, that is added to the current joint vector $\mathbf{q}$.

6.4.6 Preparing Orientational Data

In this section, the calculations for preparing the orientational data acquired from the MARG-sensor, will be presented.

Acquiring rotations relative to remote base

The MARG-sensor's internal reference for ω_z is aligned according to a magnetic field. Therefore, the raw data from the MARG-sensor's algorithm do not necessarily coincide with the *remote base* coordinate system depicted in figure 6.2. Therefore, the offset received from the Remote Operator Interface (section 6.3.3), ω_{zo}, has to be used. The following indexes will be used:

- **rb** - remote base coordinate system.
- **ms** - MARG-sensor coordinate system.

Giving orientation in remote base, $\mathbf{R}_{rb}$ by:

$$\mathbf{R}_{rb} = \mathbf{R}_{z,\omega_{zo}} \mathbf{R}_{ms} \tag{6.10}$$

where $\mathbf{R}_{z,\omega_{zo}}$ is a 3x3 dimensional basic rotation matrix for rotation around z, with the angle of the yaw offset, ω_{zo} (see appendix B for the basic rotation matrices).

Preparing for coupling

The coupling between the remote- and robot base coordinate system, was discussed in section 5.2. The coupling reference frame has a z axis that coincides with the base' z-axis. It is however rotated around z to give an x-axis that points radially out from the base towards the robot's wrist. To prepare for coupling, the forward kinematics to the wrist at q_5 is calculated. From the obtained transform, $\mathbf{T}_{wrist}^{base}$, the wrists translation vector, $\mathbf{p}_w(t)$, is extracted, where:

$$\mathbf{p}_w(t) = [x_w(t), y_w(t), z_w(t)]^T$$

Thus, the angle, ϕ_c, to rotate the remote base coordinate system about its z-axis to align it with the coupling coordinate system can be obtained by:

$$\phi_c(t) = atan2(x_w(t), y_w(t)) \tag{6.11}$$

Consequently, acquiring orientations expressed in the coupling coordinate system, **R** becomes:

$$\mathbf{R} = \mathbf{R}_{z,\phi_c(t=t_s)}\mathbf{R}_{rb} \tag{6.12}$$

where $\mathbf{R}_{z,\phi_c(t=t_s)}$ is a 3x3 dimensional basic rotation matrix for rotation around z with the angle to the coupling, ϕ_c, at the time of switch, t_s.

6.4.7 Preparing Positional Data

In this section, what has been done to prepare of the positional data acquired from the Kinect will be presented.

Filtering
Due to the the random errors in depth readings from the Kinect's raw data described in section 6.3.1, and additional noise due to tracking algorithm uncertainty a point-by-point mean-filter was used for filtering out the noise. The sample length of the filter was set to 10, hence reducing the noise of a factor of 10. However, as the code runs at a constant 100Hz, the residuals are present in the filter for $0.1s$. Therefore, the filter is reinitialized every time the main switch on the gripper analog is pushed to ensure that there are no residuals causing unwanted contributions to the control.

Acquiring position in remote base coordinates
As the Kinect is situated as depicted in figure 6.2, its coordinates has to be transformed to give positions relative to the remote base' coordinate system. The Kinect is mounted upside down, at an angle of 28° (for its z-axis to become parallel, but pointing in opposite direction, to the remote base's x-axis). As only delta values are to be used in the control, the approach used for obtaining positional values in the remote base coordinate system was:

$$x_k = y_o * cos(\alpha) - z_o * sin(\alpha) \tag{6.13}$$

$$y_k = x_o \tag{6.14}$$

$$z_k = -z_o * cos(\alpha) - y_o * sin(\alpha) \tag{6.15}$$

where α is the Kinect's angle (28°) and x_o, y_o, z_o are the filtered tracking coordinates in the Kinect's original coordinate system and x_k, y_k, z_k the tracking coordinates along the remote base coordinate system.

Acquiring the gripper analog's TCP using the MARG-sensor's orientation
The tracking algorithm for obtaining position using the Kinect, usually tracks a point as situated according to the origin of coordinate system 1 depicted in figure 6.15.

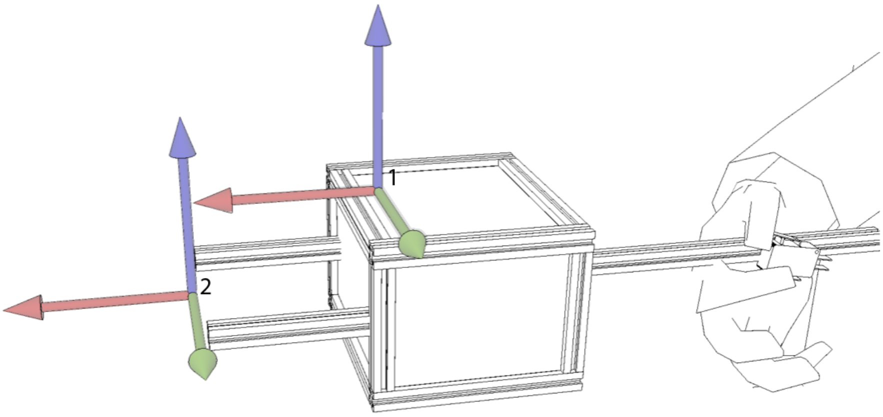


Figure 6.15: Illustration showing local coordinate systems on the gripper analog.

For obtaining the gripper analog's TCP, the orientation obtained from the MARG-sensor is utilized.
The following notation is used:
$\mathbf{r}_k = [x_k, y_k, z_k]^T$ is the three dimensional position vector from the Kinect tracking in remote base coordinates.
$\mathbf{d} = [d_x, d_y, d_z]^T$ is a three dimensional vector with the translational offset from coordinate system 1 to 2 (figure 6.15).
$\mathbf{r}_{rb} = [x_{rb}, y_{rb}, z_{rb}]^T$ is the three dimensional position vector of the gripper analog's TCP in remote base coordinates.

Calculation of the position, $\mathbf{r}_{at}$, then becomes:

$$\mathbf{r}_{rb} = \mathbf{r}_k + \mathbf{R}_{rb}\mathbf{d} \tag{6.16}$$

where $\mathbf{R}_{rb}$ is the 3x3 dimensional rotation matrix for the MARG-sensor in the remote base coordinate system.

The $\mathbf{d}$-vector was established by manual measurement and determined to be approximately $\mathbf{d} = [0.15, 0, -0.04]$.

Scaling
As presented in chapter 5, three scaling presets are available: 1:1, 1:2 and 1:3, which result in the scaled translation vector, $\mathbf{r}_s$:

$$\mathbf{r}_s = \frac{\mathbf{r}_{rb}}{s} \tag{6.17}$$

where s is the active scaling value of 1,2 or 3.

Preparing for coupling
Analogous with the preparation for coupling of the orientation, the same process has to be applied to the obtained position. By using the same rotation matrix, $\mathbf{R}_{z,\phi_c(t=t_s)}$, the position related to the robot's base coordinate system becomes:

$$\mathbf{r} = \mathbf{R}_{z,\phi_c(t=t_s)}\mathbf{r}_s \tag{6.18}$$

6.4.8 Speed Limiter and Euler Filter

Parts pertaining to the actual speed limiting in this subsection, is based on previous work by the candidate and a fellow student in a specialization project (Reme and Sanderud, 2011).

This section will present how a speed limiter and a supporting euler filter have been implemented in the remote operation system. The reasoning for implementing a speed limiter into the loop is two folded:

- Providing a layer of safety and convenience during remote operation of the industrial robot.
- Circumvent limitations of the Olimex interface.

The Olimex interface runs at approx. 100Hz. However, if the Olimex supplies a too big delta to the AX20's servo board relative to its current position, the AX20 enters a failure mode to protect the servos from jumps caused by discontinuities in the data. While a joint velocity limiter would be the most appropriate for solving this problem, the speed limiter used for providing safety in regard to tracking jumps, also removed the problems related to the Olimex when limiting to reasonable remote operation speeds.

Speed Limiter
The speed limiter follows the algorithm shown in figure 6.16. Where $\Delta P(k)$ denotes the change in suggested position at iteration k. $\Delta P(k)$ is given by

$$\Delta P(k) = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}. \tag{6.19}$$

Similarly, $\Delta O(k)$ is the change in suggested orientation at iteration k. The calculation of $\Delta O(k)$ is based on the arc lengths of the rotation, relative to TCP. It is therefore given by equation 6.20, where T_i denotes the tools length in direction i.

$$\Delta O(k) = \Delta A T_x + \Delta B T_y + \Delta C T_z. \tag{6.20}$$

$\Delta M(k)$ denotes the longest movement, of either position or orientation, thereby simply the greater of $\Delta P(k)$ and $\Delta O(k)$.

$$\Delta M(k) = max(\Delta P(k), \Delta O(k)) \tag{6.21}$$

$\Delta M(k)$ is referenced against the maximum allowed movement, ΔM_{max}, which is given by:

$$\Delta M_{max} = \frac{v_{max}}{f_0} \tag{6.22}$$

where v_{max} denotes the maximum allowed speed, and f_0 is the frequency of the control system.

Then, if $\Delta M(k)$ is greater than ΔM_{max}, the contribution is limited at iteration k. So that the output of the limiter becomes:

$$\Delta P = \Delta P \frac{\Delta M_{max}}{\Delta M} \tag{6.23}$$

$$\Delta O = \Delta O \frac{\Delta M_{max}}{\Delta M} \tag{6.24}$$

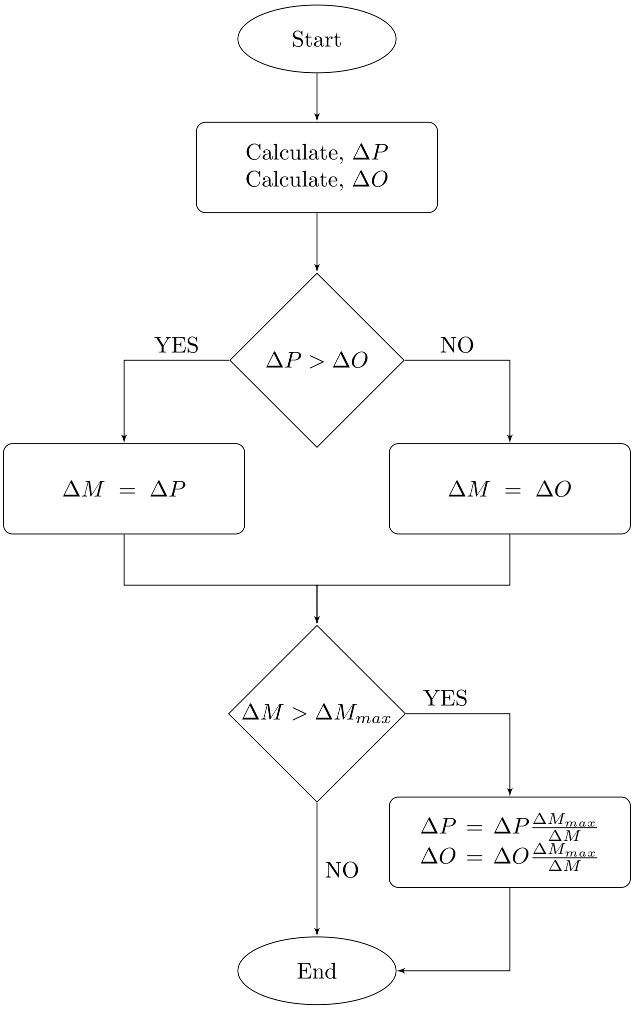


Figure 6.16: The algorithm governing speed limiter.

Euler filter

The change in orientation may exceed 180°, because delta values that are to be speed limited are relative to the point in time of which the main switch was activated. The algorithm used for speed limiting uses euler angles, which has discontinuities where ω_x and ω_z jumps from $\pm 180° \rightarrow \mp 180°$. This can cause the speed limiting algorithm to choose unreasonable trajectories for reaching the desired orientation. To remedy this, an euler filter is implemented as in figure 6.17.

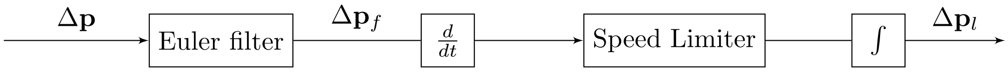


Figure 6.17: Overview of the preparation process for the speed limiting.

The euler filter extracts the yaw and roll from the 6 dimensional $\Delta\mathbf{p}$-vector. The discontinuities in yaw and roll are subsequently patched and reinserted into the vector, giving the $\Delta\mathbf{p}_f$-vector. Figure 6.18 shows how the patching is performed, where θ_i is ω_x or ω_z and θ_{f_i} is the output from the filtering algorithm for the respective angle.

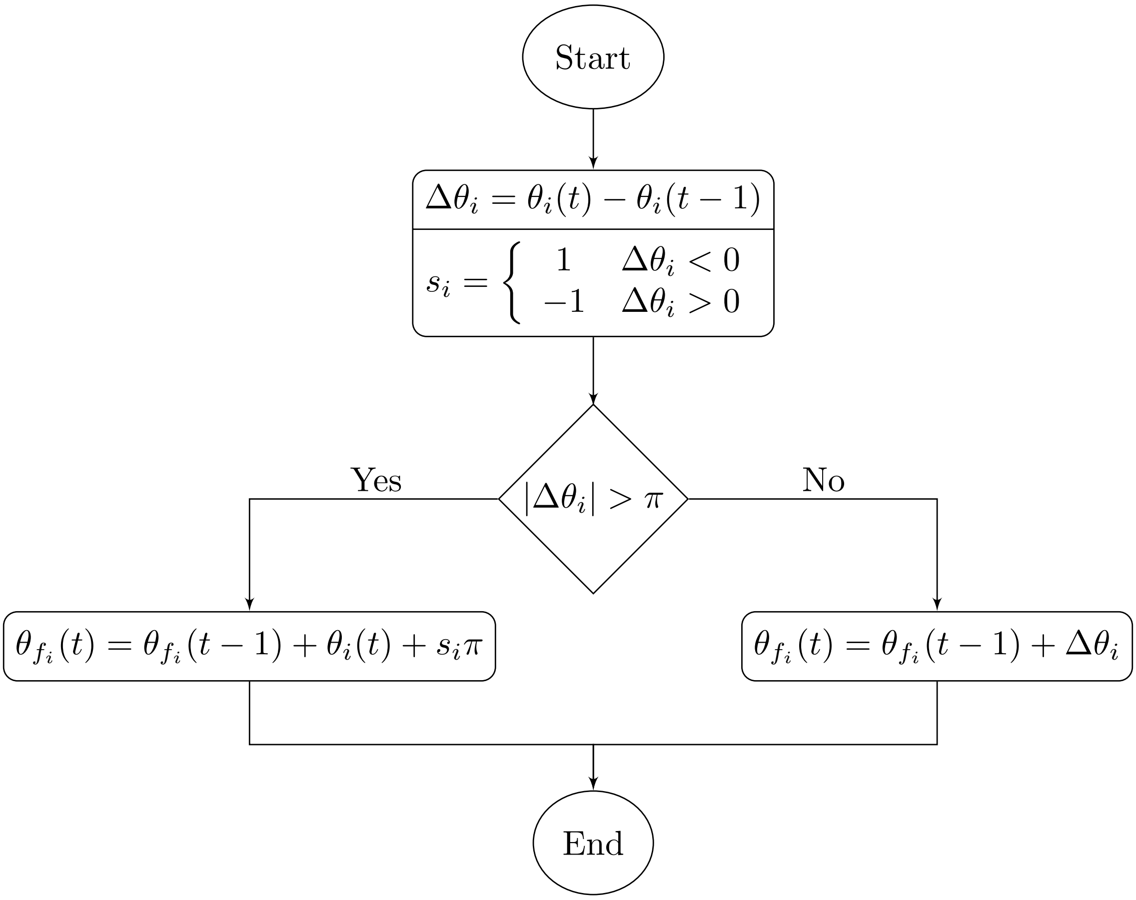


Figure 6.18: Euler filter algorithm

Figure 6.19 show a discontinuity in ω_z patched by the algorithm.

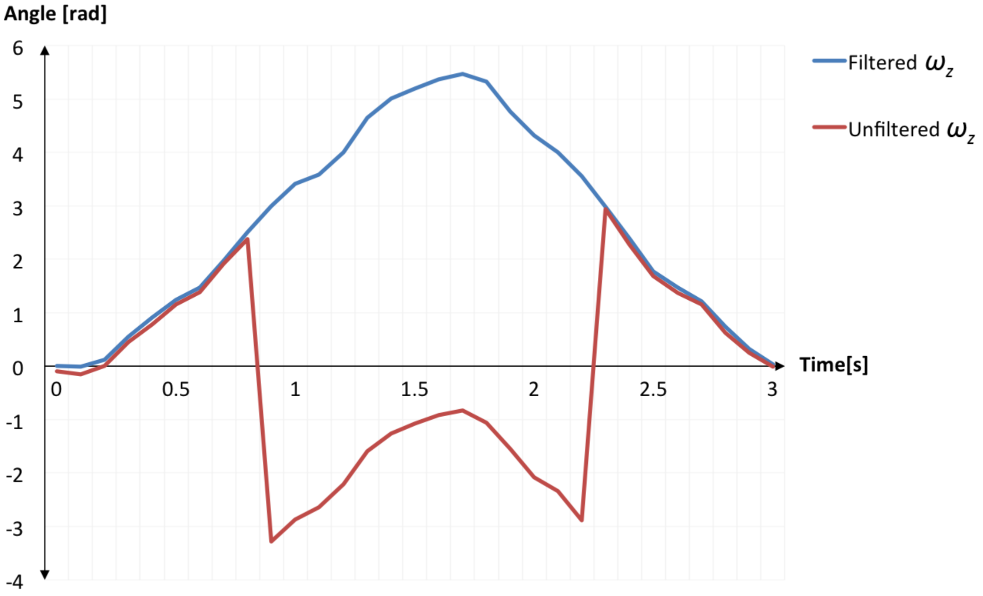


Figure 6.19: Result of filtering for ω_z

6.4.9 Robot Program Creation

The implementation allows for creation of simple robot programs that can be exported to the NACHI AX20. It uses NACHI's *MOVEX*-command, which allows for programming positions based on pose-vectors, joint angles or encoder values. An example of a MOVEX command-string using joint-angles or encoder values is:

```
MOVEX A=7,AC=0,SM=0,M1E,L,(q2, q3, q4, q5, q6, q7),S=300,H=1,
MS,M2E,P,(q1),S=300,H=1
```

Where `AC` is acceleration, `SM` is smoothness, `S` is speed, and `L` or `P` in front of the joints denotes linear interpolation or joint move respectively. And $q_1 \rightarrow q_7$ are joint-angles in radians or encoder values in hexadecimals.

In the implementation, points are saved if the operator commands this from the external controller in his hand or via *Programming and motion guidance settings*-interface. A MOVEX command is then created based on the current servo encoder values acquired from the Olimex, and the selected settings for interpolation, smoothness, acceleration and speed from the interface. The result is then appended into a list view, and after successfully programming the desired points, the program can be saved to a file. A screenshot of the *Programming and motion guidance settings*-interface can be seen in figure 7.3 in section 7.5.

Chapter 7

User Guide

Most details regarding the practical usage of the system is considered sufficiently presented in chapter 5. Consequently, this chapter is intended to serve as a guide for startup and initial checks associated with using the Cognitive Remote Operation System. In presenting the startup process, it is distinguished between the *workstation* in the remote room and the *server* in the local room. When referring to the server, it is meant that the instructions are to be made via e.g. *remote desktop services* to the server.

To connect to the server via remote desktop services, click the icon on the workstation desktop called *Client-Server*. If manual configuration is preferred, the necessary information is as follows:

IP: 192.168.212.108

Username: Administrator

Password: 123

7.1 3D-Simulation and Animation

Starting the 3D-simulation and animation modules from the PPM's original CROP system is done as follows: (*This has to be done in the presented order.*)

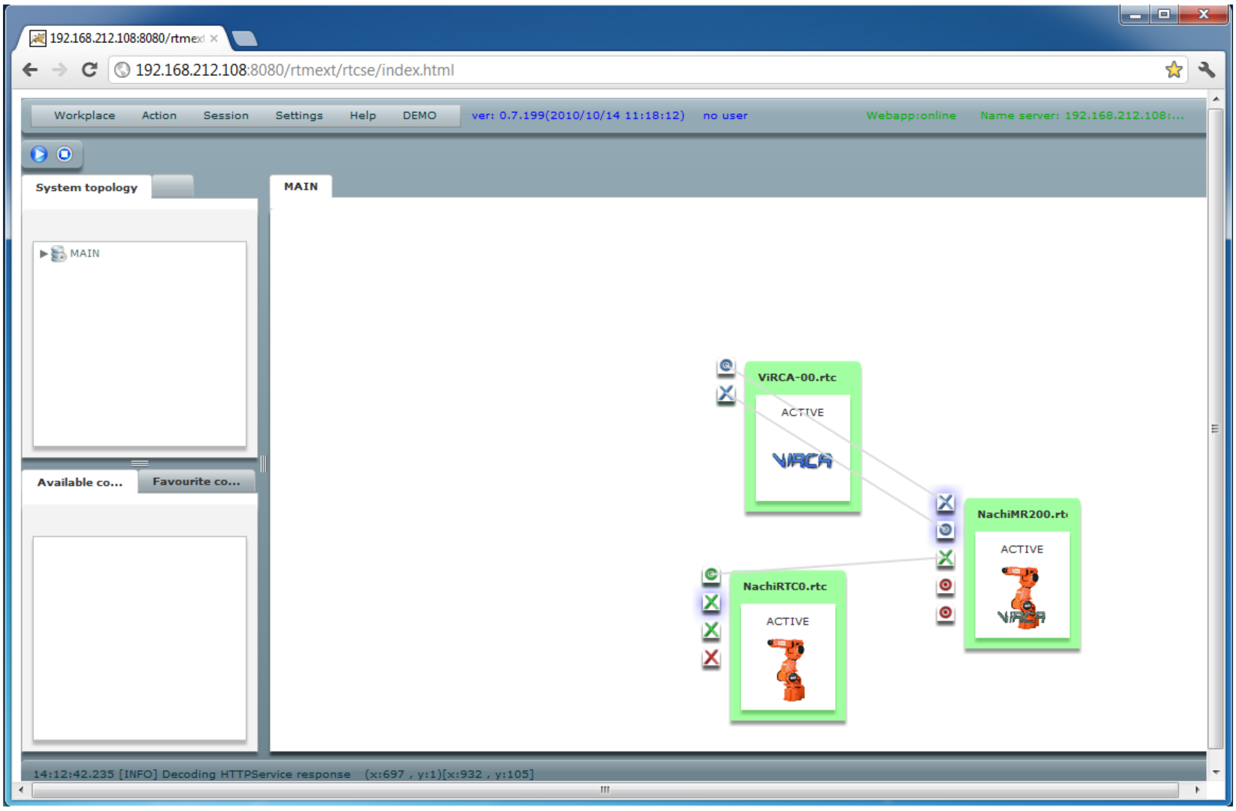
On the server:

- Click the desktop icon named *omni*.
- Click the desktop icon named *join ui*.
- Click the desktop icon named *NachiMR20CyberDevice*.
- Click the desktop icon named *NachiRemoteServer*.

On the workstation:
Start by clicking the desktop icon named *VirCA*. As VirCA is successfully started, open a web browser and go to the following url: http://192.168.212.108:8080/rtmext/rtcse/index.html (this has been set as a bookmark in the Chrome browser on the workstation). In the web interface, do the following:

- Click on *Demo*.
- In the upper left, click *Workplace* followed by *Load from local storage*.
- In the list view, choose the *VircaServer* preset (second from the top), followed by clicking *Load*.
- In the new dialog, click *Play*.

At this point, all the RTC components should be lit green as depicted in figure 7.1. In addition, a 3D-model of the NACHI MR20 should appear in VirCA (pointing into the ground due to no data present yet).



```
14:12:42.235 [INFO] Decoding HTTPService response   (x:697 , y:1)[x:932 , y:105]
```

Figure 7.1: Screenshot web interface with correctly lit components.

7.2 NACHI AX20

Before one can turn on the NACHI AX20 controller the following has to be checked:

1. That the Olimex interface is correctly connected between the controller- and servo board.
2. That the Olimex has power.
3. That the Olimex is set up to intercept signals as well as receiving and broadcasting. This is done by connecting to it at IP: 192.168.212.30 via SSH. Subsequently logging in as "*root*" with the password "*olimex*". Then type the command for starting the process: "*nasta*".

At this point, the NACHI AX20 can be powered on.

When the NACHI AX20 is finished with its startup process, robot program 84 could be run in continuous playback mode for giving suitable start-configuration of the robot (this program is included in the digital delivery, see **APPENDIX**). However, this is not a prerequisite for testing the system. It only provides an initial position away from a singularity.

7.3 Remote Operation Server

The Remote Operation Server is a LabVIEW application that is intended to run on the local server, where all calculations pertaining to the manipulation of the robot is performed. The startup procedure of the Remote Operation Server is as follows:

On the server:

- Click the *Remote Operation Server*-folder on the desktop.
- Launch the *FrontPanel.vi*.

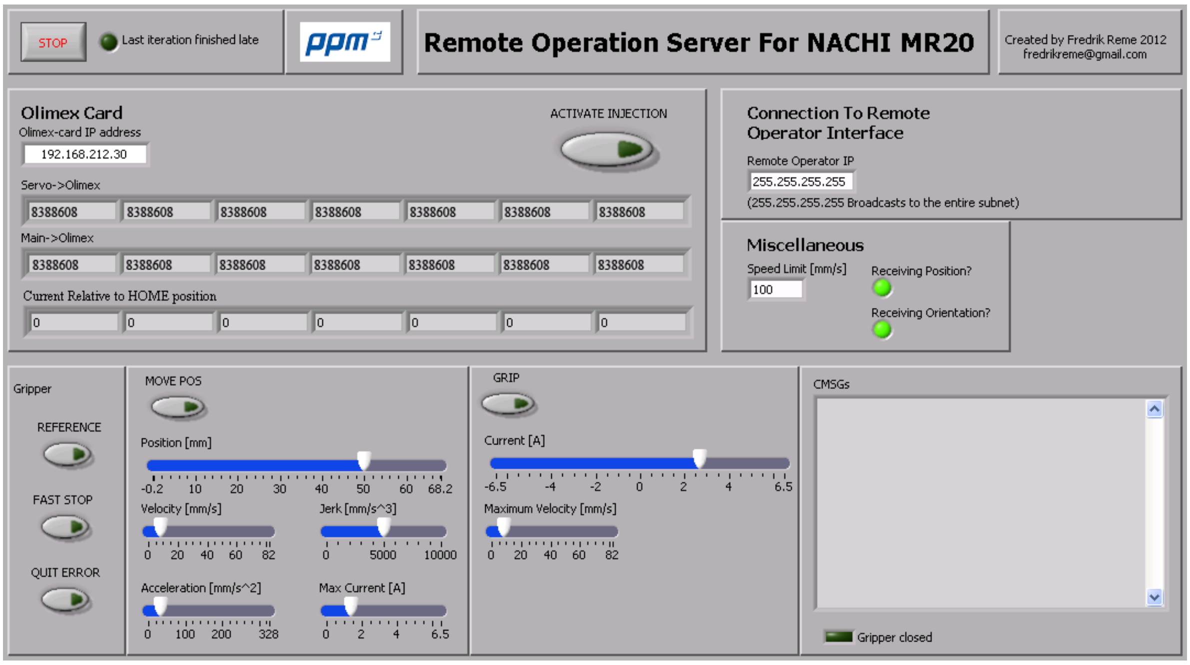
Two VI-Panels will be displayed, the *FrontPanel.vi* and *makerobotprog.vi*. The FrontPanel.vi is the one of interest in starting the remote operation system, while the makerobotprog.vi is intended to be remotely connected controlled from the remote workstation so that the operator can customize programming parameters and motion guidance settings.

Figure 7.2 depicts the FrontPanel.vi. If remote operation is to be done from outside PPM's laboratories, the remote workstation should connect via VPN and its IP should be inserted into the "Remote Operator IP"-textfield[1]. In the panel, also the speedlimit during remote operation can be set, in addition to indicators for the connection to the *Remote Operator Interface*.

---

[1] This has not been tested.

The *Activate Injection*-button makes the application send its contributions to the Olimex for injection in joint-space, and can be considered an ON/OFF button of the Remote Operation Server. However, before injection is started, one should verify that the *Servo→Olimex* and *Main→Olimex* values are in the vicinity of 8 300 000. Any deviations from this could mean that the AX20 is not running or that the Olimex is not working correctly.


Remote Operation Server For NACHI MR20

Created by Fredrik Reme 2012
fredrikreme@gmail.com

Olimex Card
Olimex-card IP address: 192.168.212.30
Servo->Olimex: 8388608 8388608 8388608 8388608 8388608 8388608 8388608
Main->Olimex: 8388608 8388608 8388608 8388608 8388608 8388608 8388608
Current Relative to HOME position: 0 0 0 0 0 0 0

Connection To Remote Operator Interface
Remote Operator IP: 255.255.255.255
(255.255.255.255 Broadcasts to the entire subnet)

Miscellaneous
Speed Limit [mm/s]: 100

Figure 7.2: Screenshot of the FrontPanel.vi of the Remote Operation Server.

7.4 Remote Operator Interface

The Remote Operator Interface is intended to run on the remote workstation. The following should be verified before starting it:

- That the Microsoft Kinect is powered up and connected to the computer via USB.
- That the gripper analog is connected to the computer via USB.

If this is true, it can be launched by clicking the *RemoteOperatorInterface*-icon on the workstation's desktop.

The Remote Operator Interface has been depicted previously in figure 6.5. Before initiating control, the user should verify that the offset in ω_z of the gripper analog is correctly set. If the offset is wrong, it is set by pointing the gripper analog directly towards the computer monitor and hitting the *a*-button on the workstations keyboard. Lastly, for the position tracking the gripper analog using the Kinect, the gripper analog should be waved back-and-forth. When a tracking point is acquired, the point will be displayed as a red dot in the Kinect's depth-map view.

7.5 Programming and Motion Guidance Settings

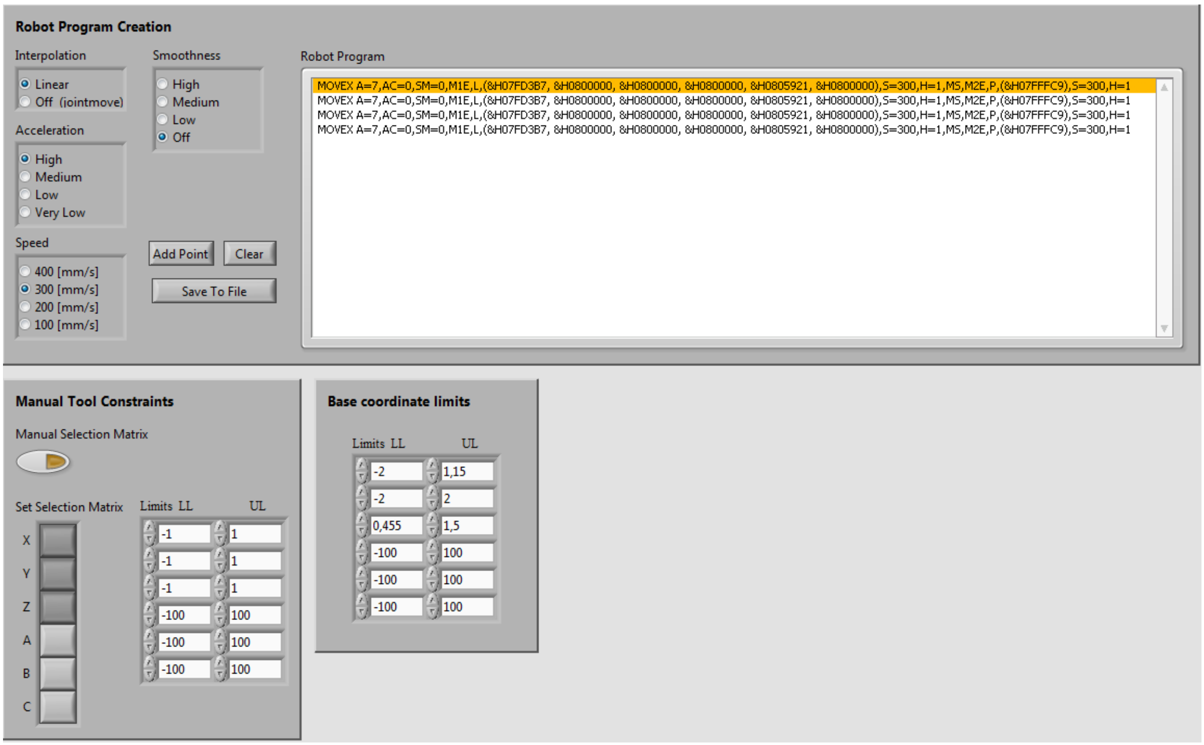
As presented in section 7.3, the programming and motion guidance settings interface actually run on the server. The intention is to remotely connect to this panel by using LabVIEW's remote panel function.

On the workstation:

- Start LabVIEW 2011
- Click on *Operate* followed by *Connect to Remote Panel*.
- In the dialog box, use IP: 192.168.212.108, Port: 8080 and VI Name: ROS.lvproj/My Computer/makerobotprog.vi
- Check *Request control*, and hit *Connect*.

Figure 7.3 depicts the programming and motion guidance settings. In this interface, the operator change the various parameters that is to be applied when either the *Add point*-button or the physical button on the operator's external controller is pushed. The program can be be saved by hitting the *Save to File*-button, which saves the program to \ \ PPM_Extra \ media \ students \.

In addition, a manual selection matrix can be set in the lower part of the interface. This is done by latching the buttons corresponding to the directions that is to be allowed. The tool coordinate constraints are specified in the left *Limits*-table and base coordinate limits are specified in the *Limits*-table to the right. All units are in meters.



Interpolation
● Linear
○ Off (jointmove)

Smoothness
○ High
○ Medium
○ Low
● Off

Acceleration
● High
○ Medium
○ Low
○ Very Low

Speed
○ 400 [mm/s]
● 300 [mm/s]
○ 200 [mm/s]
○ 100 [mm/s]

Robot Program

```
MOVEX A=7,AC=0,SM=0,M1E,L,(&H07FD3B7, &H0800000, &H0800000, &H0800000, &H0805921, &H0800000),S=300,H=1,MS,M2E,P,(&H07FFFC9),S=300,H=1
MOVEX A=7,AC=0,SM=0,M1E,L,(&H07FD3B7, &H0800000, &H0800000, &H0800000, &H0805921, &H0800000),S=300,H=1,MS,M2E,P,(&H07FFFC9),S=300,H=1
MOVEX A=7,AC=0,SM=0,M1E,L,(&H07FD3B7, &H0800000, &H0800000, &H0800000, &H0805921, &H0800000),S=300,H=1,MS,M2E,P,(&H07FFFC9),S=300,H=1
MOVEX A=7,AC=0,SM=0,M1E,L,(&H07FD3B7, &H0800000, &H0800000, &H0800000, &H0805921, &H0800000),S=300,H=1,MS,M2E,P,(&H07FFFC9),S=300,H=1
```

Limits LL	UL
-1	1
-1	1
-1	1
-100	100
-100	100
-100	100

Limits LL	UL
-2	1,15
-2	2
0,455	1,5
-100	100
-100	100
-100	100

Figure 7.3: Screenshot of the programming and motion guidance settings.

Chapter 8

Experiments, Tests and Results

The following chapter will present the experiments that were carried out and the results they gave. The chapter is divided in three:

- System function (section 8.1)
- Time-delay (section 8.2)
- Usability tests (section 8.3)

All experiments were run with the following settings: Max speed = $100 mm/s$ and limits in robot base coordinates: Lower limit on z-value = $0.465 m$ (approx. height of the table).

The core layout of the robot's- and operator's location during most experiments was as depicted in figure 8.1.

Figure 8.1: Main layout during the experiments.

8.1 System Function

The system function experiments are intended to indicate how the system performs while using the various system functions. All experiments have been video documented, and should be considered a part of the result.

8.1.1 Scaling

This experiment demonstrates the translational scaling available to the operator. During the experiment two identical tables were placed at both the operator's and robot's location. Marked on the robot's table, are two 2x2 grids as shown in figure 8.2, while at the operator's table, only the big, outer grid is marked.

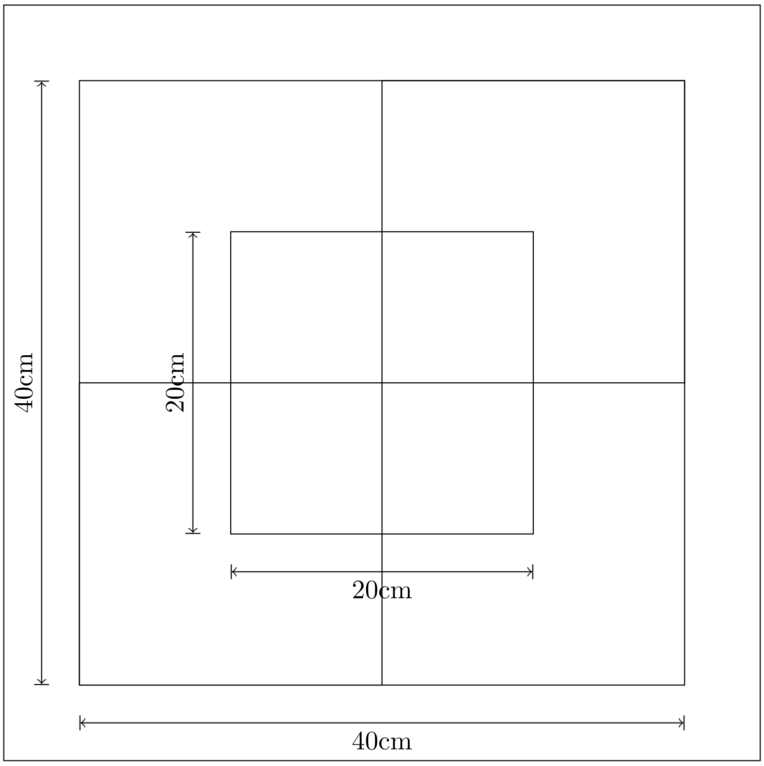


Figure 8.2: Top-view illustration of the table grid with its dimensions.

During the experiment, the table grid is used by the operator as a reference. In figure 8.3, one can see the gripper analogs position together with the resulting positions of the robot in 1:1 scaling for reference, and for 1:2 scaling. The experiment can be seen in its entirety in video 1.

The chart in figure 8.4 shows an excerpt from the experiment while utilizing 1:2 scaling. As can be seen in this chart, the Δy is scaled properly, and the the robot assumes correct position. Between $4.5s$ and $6.5s$, one can observe areas where the Δy for the scaled values are steeper than for the values returned from the MR20's servos (blue). This can be attributed to the speed limiter which was set to $100mm/s$.

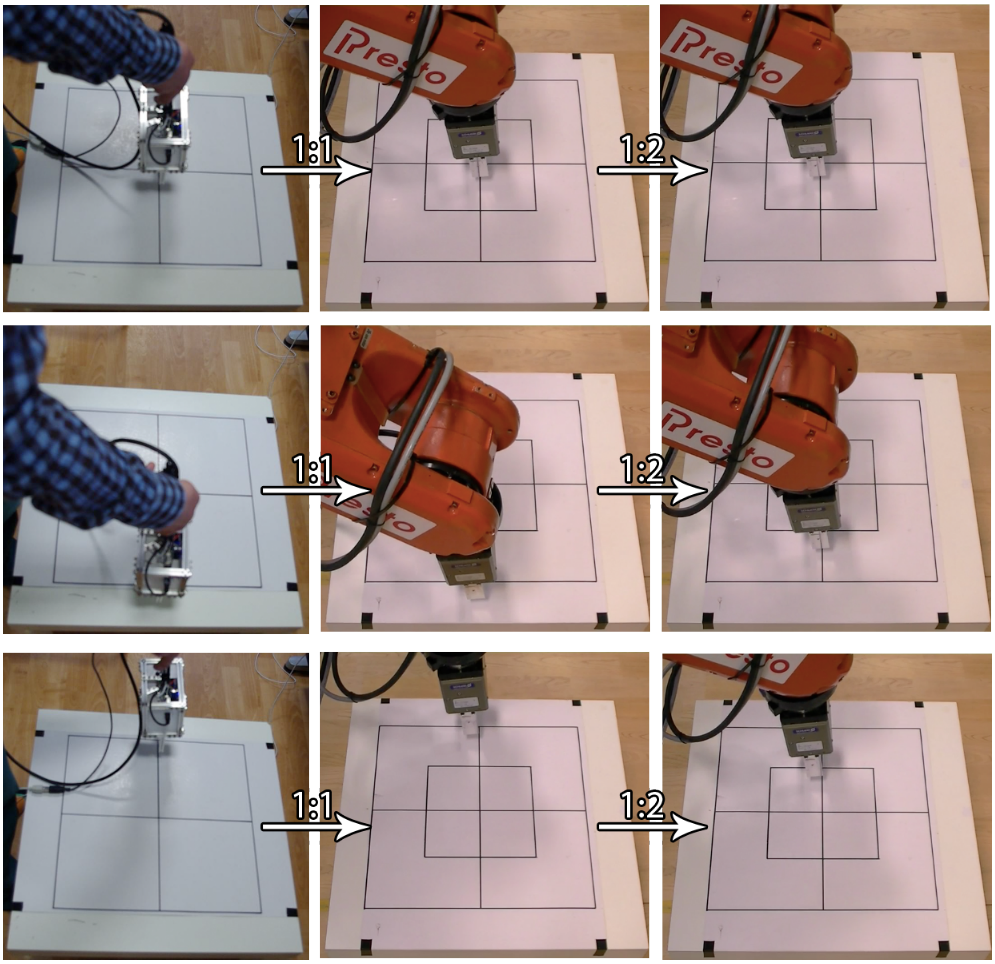


Figure 8.3: Gripper analog positions and corresponding result in 1:1 and 1:2-scale.

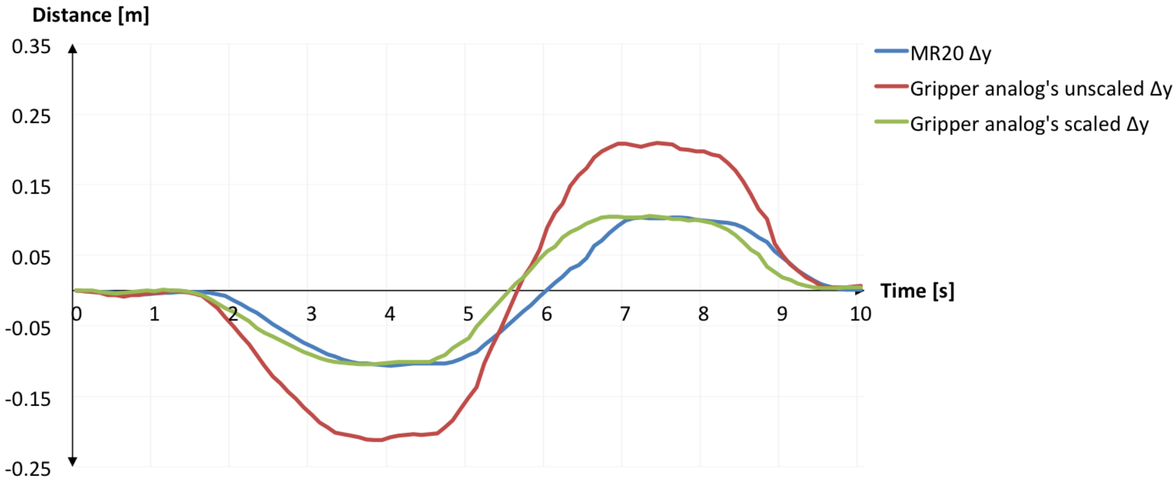


Figure 8.4: MR20's Δy plotted against the gripper analog's scaled and unscaled Δy during 1:2-scaling as shown in figure 8.3

8.1.2 Self-motion

Self-motion is an important tool for redundant robots when performing tasks in hard-to-reach locations. In this experiment, the task performed was to grip a workpiece placed parallel to an obstacle beside the robot's elbow. The workpiece was to be picked up with an appropriate orientation of the gripper, which is a task specification that a six-axis robot would not be able to fulfill under these circumstances. The experiment can be seen in its entirety in video 2.

Figure 8.5 shows frames from video 2. In frame 2, 3 and 4, one can see the grippers position and orientation stays constant while the MR20 reconfigures it's elbow in preparation for reaching the object.

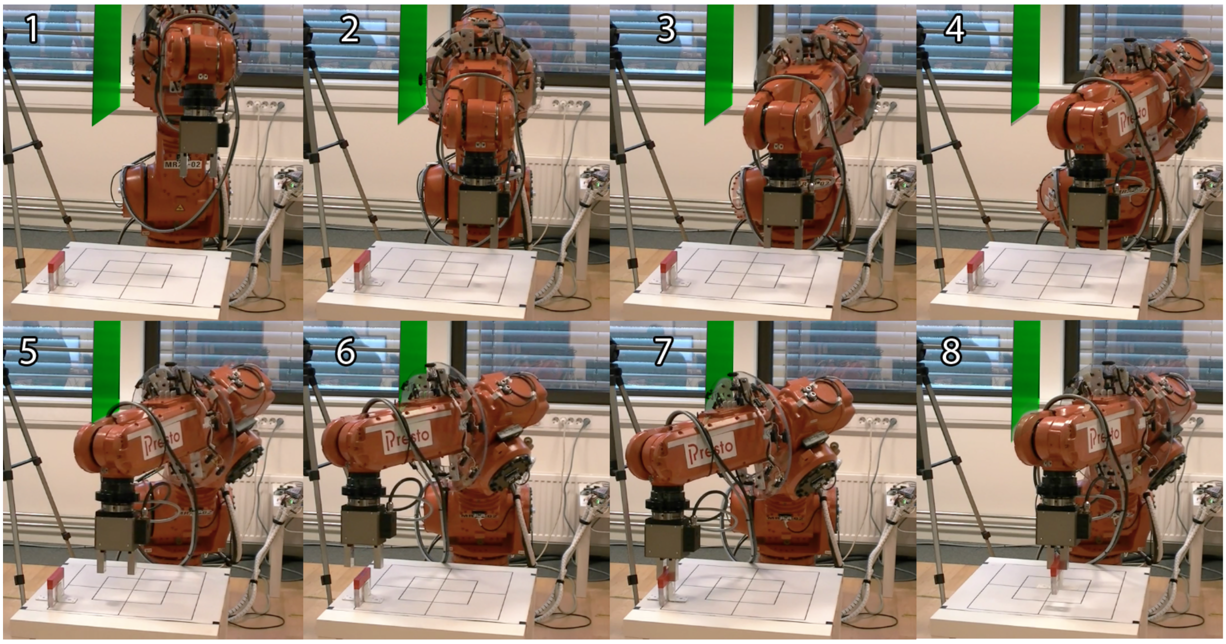


Figure 8.5: Self motion experiment. Obstacle marked with green.

The charts in figure 8.6, 8.7 and 8.8 show the robot's joint angles, position and orientation (base coordinates), respectively, during the course of the experiment. By using the frames from figure 8.5 as reference, one can see that the frames 2 → 4 corresponds to $20s-35s$. The charts in figure 8.7 and 8.8, demonstrates that neither position nor orientation changes during the self-motion.

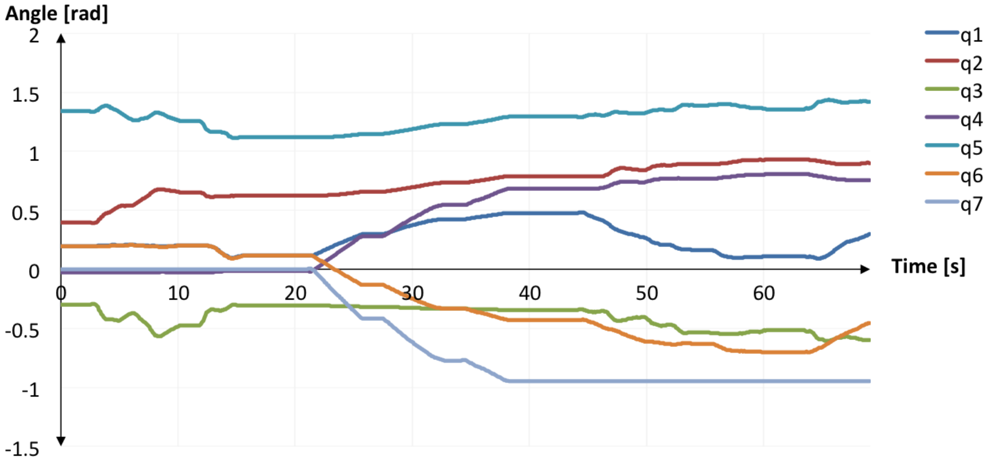


Figure 8.6: MR20's joint angles during the self-motion experiment.

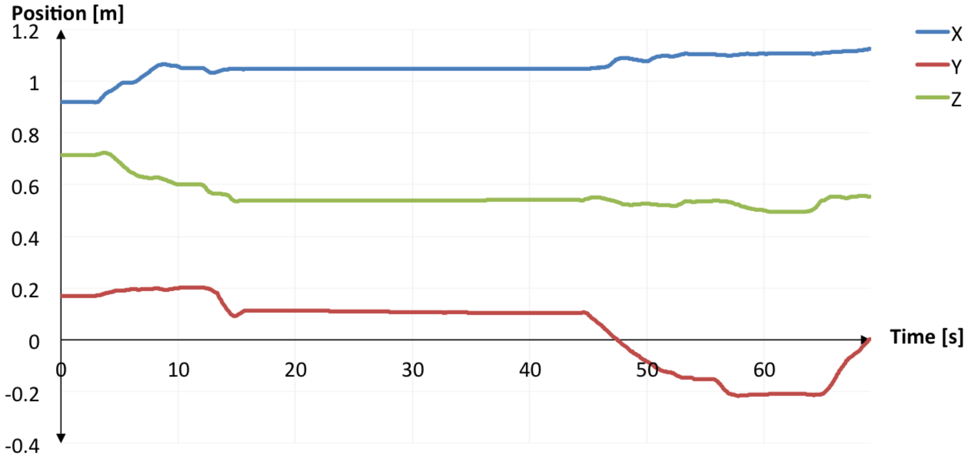


Figure 8.7: MR20's tool position in base coordinates during the self-motion experiment.

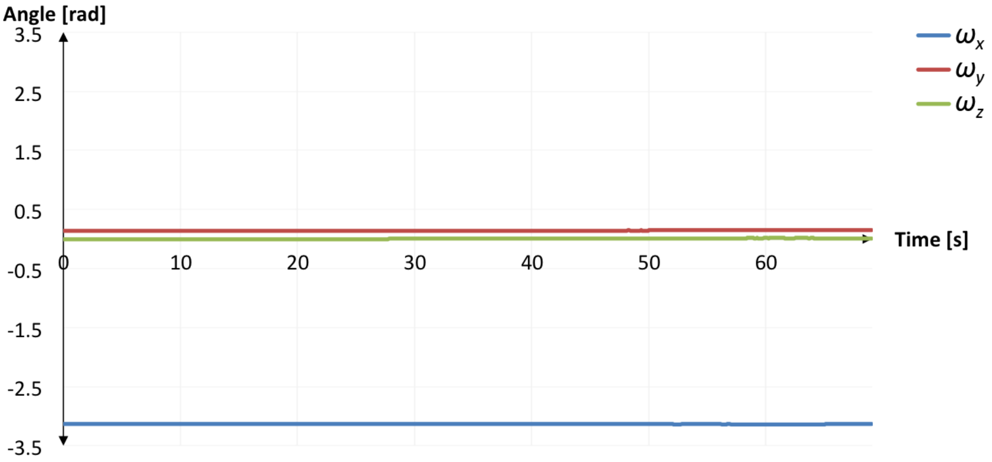


Figure 8.8: MR20's tool orientation in base coordinates during the self-motion experiment.

8.1.3 Motion Guidance

The manual constraints can be used as tool for motion guidance in unstructured environments. For testing the manual constraints, a plate was position at an arbitrary angle in the robot's workspace. In the experiment, the objective is to use the manual constraints to give guided motions over the plane. The experiment can be seen in its entirety in video 3.

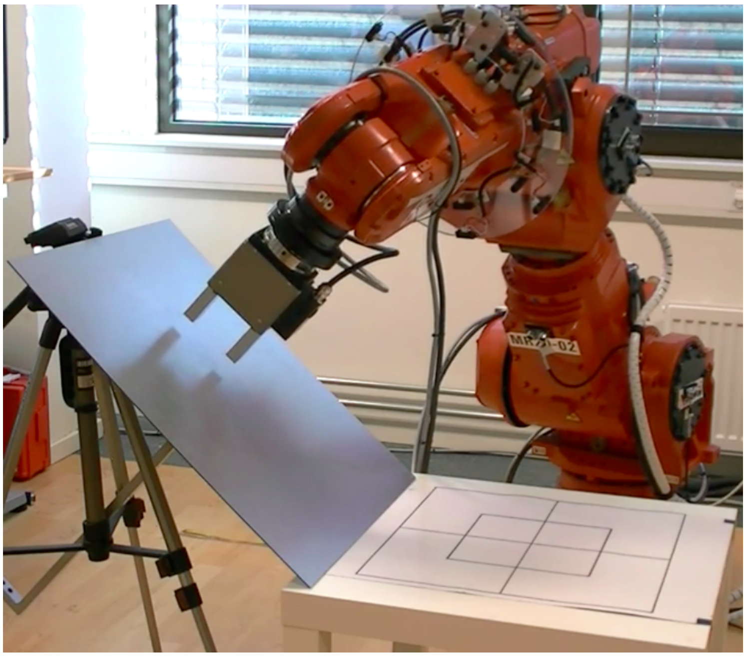


Figure 8.9: Excerpt from the experiment showing the plate placed at an arbitrary angle.

During the experiment, the robot's end-effector was placed at an appropriate pose, near perpendicular to the plate. At this point, the selection matrix 8.1 was activated with the tool coordinate constraints from table 8.1.

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{8.1}$$

	Lower limit	Upper Limit
X	$-\infty$	∞
Y	$-\infty$	∞
Z	$-\infty$	0
ω_x	$-\infty$	∞
ω_y	$-\infty$	∞
ω_z	$-\infty$	∞

Table 8.1: Table of limits in tool coordinates.

Figure 8.10, shows a chart of the unconstrained delta values from the gripper analog in tool coordinates during the experiment. Figure 8.11, shows a chart of the delta values actually applied to the robot in tool coordinates under the constraints of table 8.1.

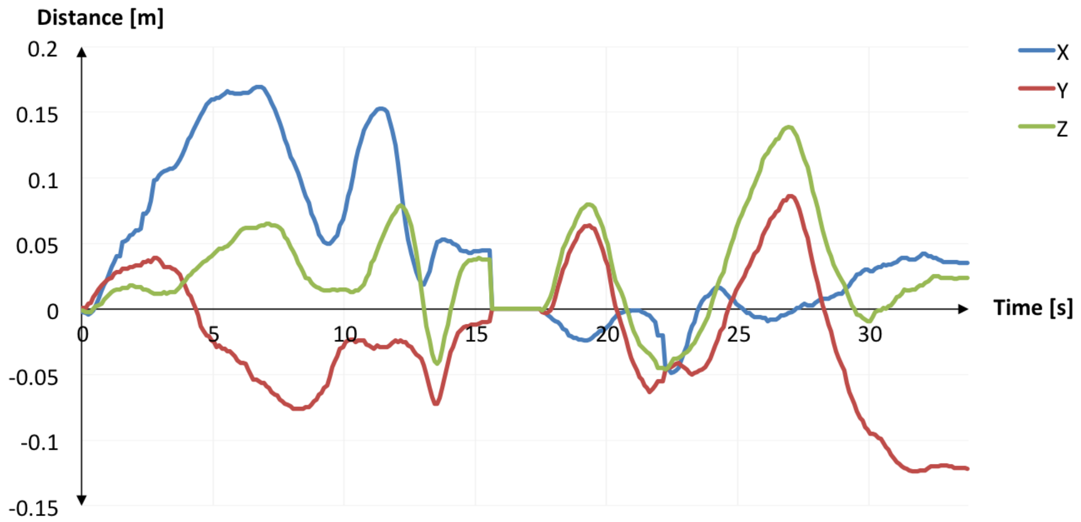


Figure 8.10: Unconstrained delta values in tool coordinates

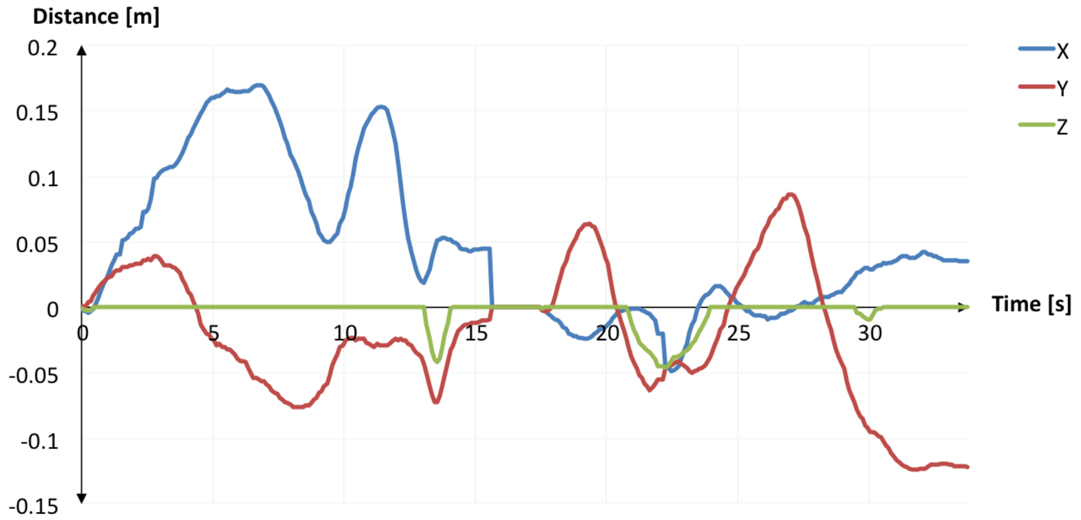


Figure 8.11: Constrained Δz values of the gripper analog in tool coordinates.

As can be seen from the figure 8.10 and 8.11, positive z-directional deltas from the gripper analog are successfully masked out, consequently constraining the robot to not move any closer to the plane than the distance from the plane during initial alignment. As negative z-movements are however allowed, the robot can move away from the plane, as can seen be at $\approx 13sec$ and $\approx 21-24sec$ in figure 8.11.

8.1.4 Robot Programming

This experiment is intended to test the robot programming functionality implemented in the system. In the experiment, the robot was moved around to seven locations where points were saved. The resulting robot program was subsequently compiled on the NACHI AX20 robot controller and played back.

The position and orientation of the tool was recorded both during the programming process and during playback of the resulting robot program. The charts in figure 8.13 and 8.14, show positions of during programming and during playback, respectively. By the same token, the charts in figure 8.15 and 8.16 show the corresponding orientation. In all charts, the saved points are indicated by vertical lines. Figure 8.12 shows the first six points programmed during the experiment.

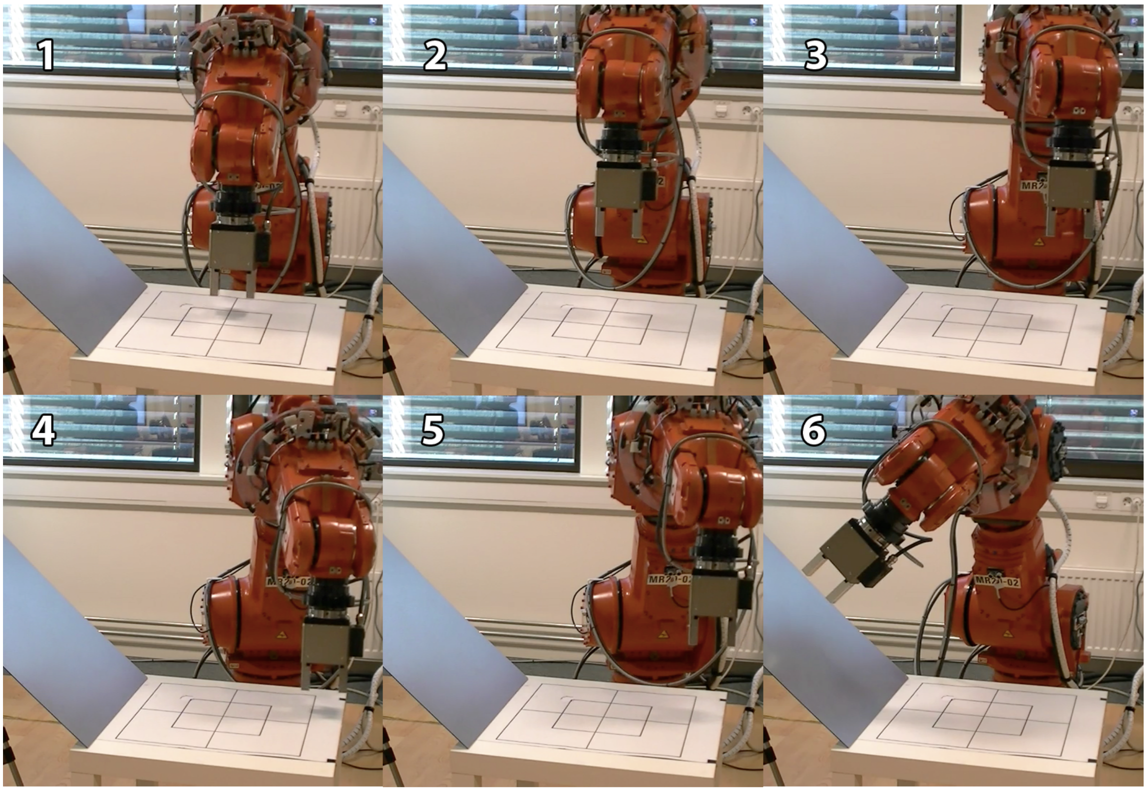


Figure 8.12: Excerpt from video 4. Six the programmed points.

The programming of the robot during the experiment, as well as the NACHI AX20 executing the program can be seen in video 4. And the resulting robot program can be found in appendix D.

As can be seen from the charts, the resulting robot program makes the robot take the correct poses. The discrepancies in the start of the charts can be attributed to the fact that the robot did not have the same initial pose when starting the programming and playback of the program on the NACHI AX20.

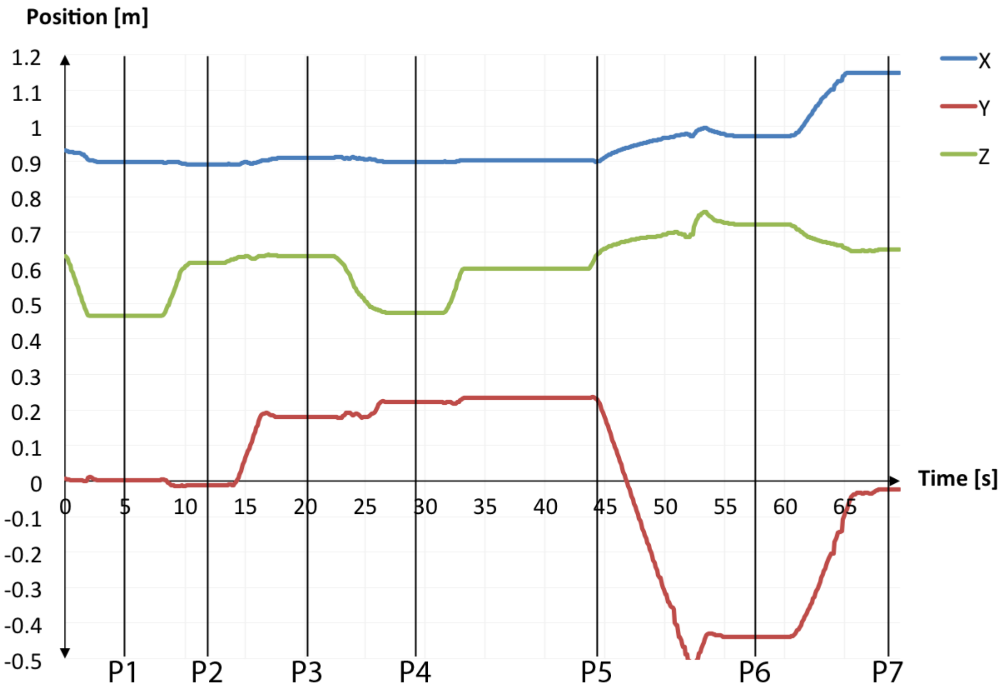


Figure 8.13: Position and saved points during programming experiment

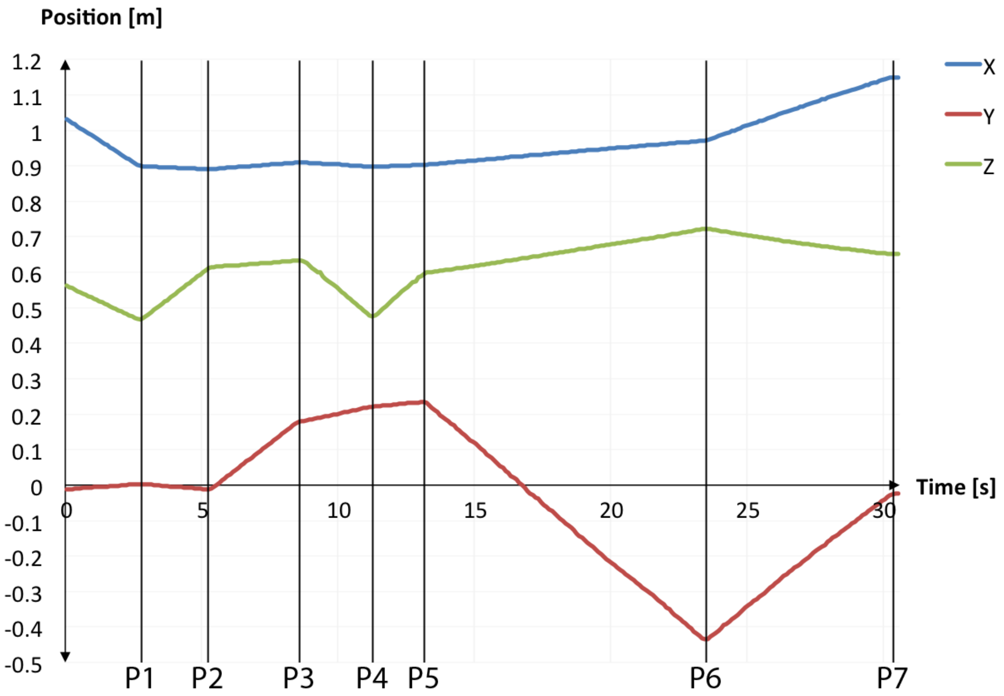


Figure 8.14: Position during NACHI AX20's playback of the resulting program.

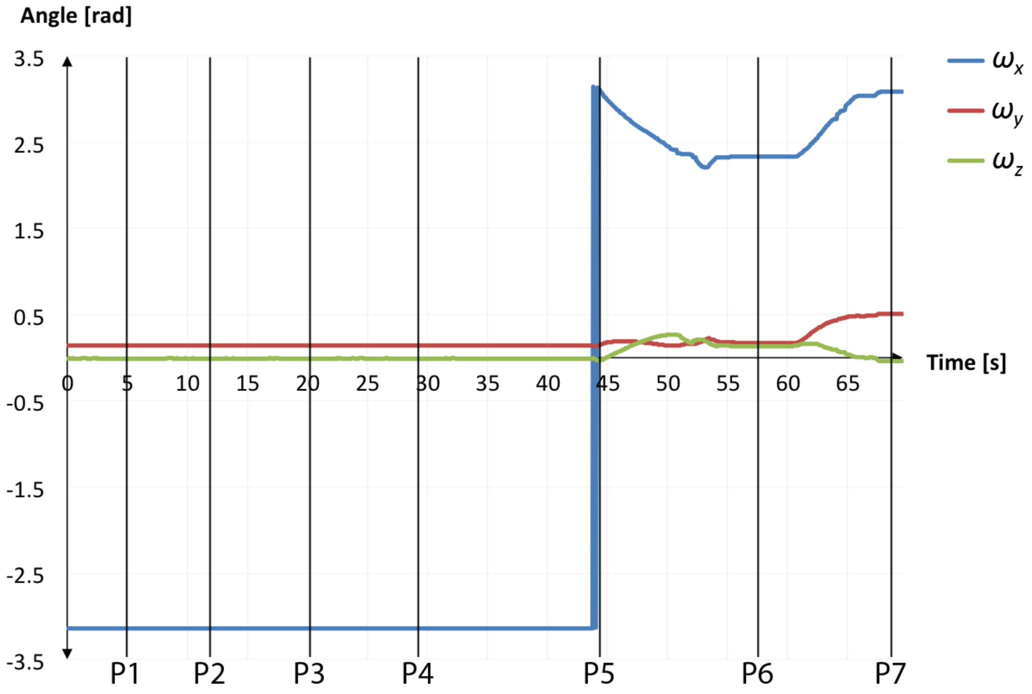


Figure 8.15: Orientation and saved points during programming experiment

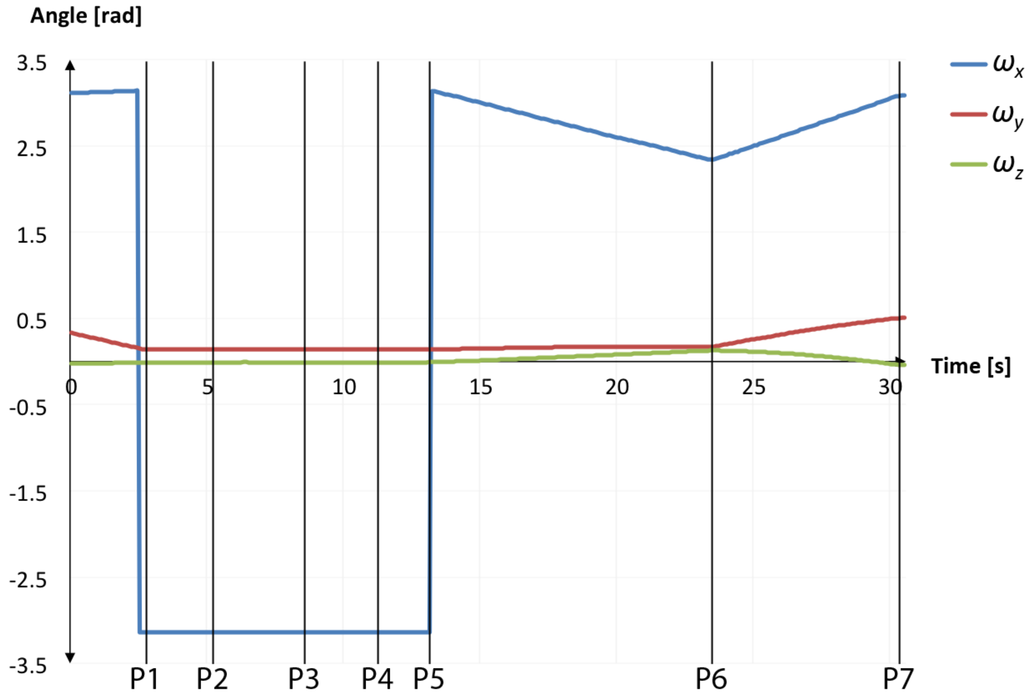


Figure 8.16: Orientation during NACHI AX20's playback of the resulting program.

8.2 Time-Delay

Time-delay is one of the most common challenges that remote operation systems have to deal with. To examine how the system performs under more realistic conditions, a time-delay of one second was introduced to the system. This was done by introducing a first-in first-out buffer (FIFO) to the outputs of the Remote Operation Server, consequently delaying the time between user input and execution by one second.

The task performed in this experiment was a pick-and-place of the workpiece depicted on the table in figure 8.17. It was to be placed on the corner of the opposite quadrant of the worktable. The experiment is evaluated based on observations from the candidate during the experiment. The experiment can be seen in its entirety in video 5.

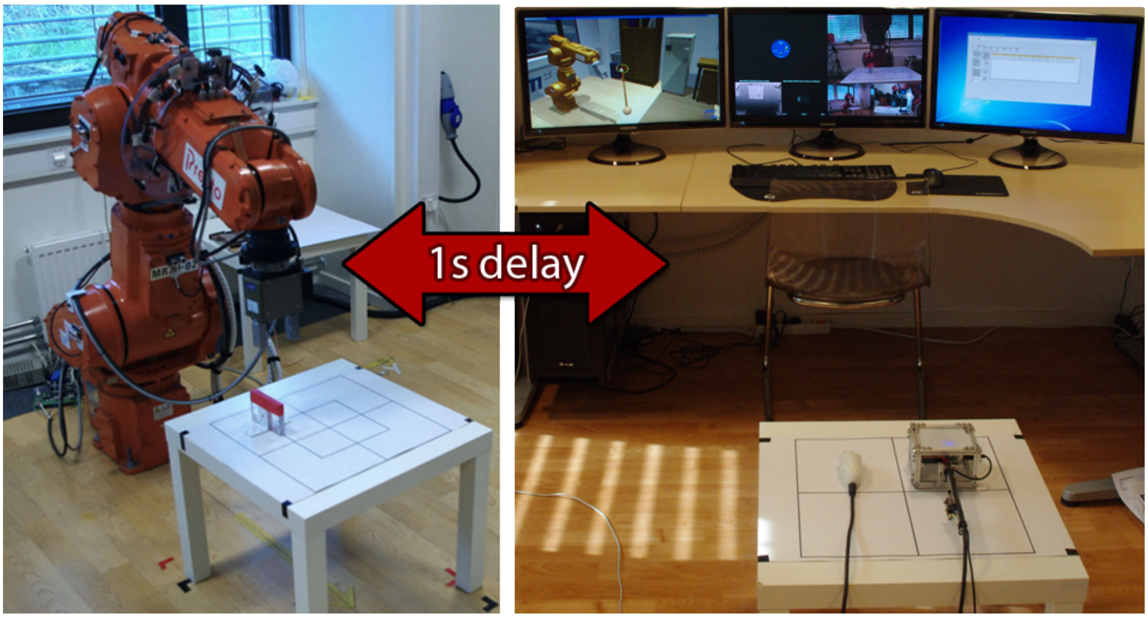


Figure 8.17: Experiment performed with 1s timedelay.

For completing the task, the time-delay was considered more of an inconvenience than a hindrance. The overall function of the system was as expected, while the workflow for completing the task changed. The time delay made it necessary to adopt a behavior dominated by a "move-and-wait"-strategy. This consequently caused the total task to be performed considerably slower than under ideal conditions.

The fact that the system uses a position-based control device, made it possible for the candidate to plan out relatively big motions by relating them to his local environment before executing them. For the initial and final approach towards the table, however, the "move-and-wait"-strategy was a necessity.

8.3 Usability Tests

Five persons from varying backgrounds tested the general usability of the system. Some of the participants were well versed in the field of robotics, while others were new to it. Thus, the test could be considered a mixture of hallway testing and expert review. The use of random people not familiar to the subject, often reveals some weaknesses that could be overlooked by expert reviewers. An expert reviewer could already have a good idea of how to accomplish the task, thereby not exploring the system in the same manner as a non-expert. Moreover, non-experts, lacking background knowledge, may not have sufficient perspective to properly analyze the system. See appendix E for a brief overview of the various participants backgrounds.

The participants were instructed to perform a simple pick-and-place task. They were given two attempts at moving a workpiece from one side of a table to the other and back. Before starting, the following initial briefing was given:

- Informing that control would be activated by holding down the main switch on the gripper analog, and by letting go the robot's movement would seize. It was also emphasized that the participant should let go of the main switch if any emergencies or other concerns arose.
- Clarifying that the perspective for control is out from the robot's base, thus the participant must consider him-/herself as the robotic arm.
- A short introduction, to non-experts, on the practical implications of the various control-modes, such as 6DOF, translation, rotation and scaling.

The main issues examined during the usability tests were:

- How does the participant respond to the screen gradually becoming red?
- Does the user seem to understand and use the button interface and it's functions?
- Did the participant reach a level of proficiency in using the system?

Before testing, the singularity warning limit for joint six (the middle joint of the wrist) was increased considerably (from 28° to 68°). This was done to increase the probability for the screen turning red during testing. In addition, for encouraging the participant to explore the control-modes, the tests started with full 6DOF and 1:1 scaling, which could be considered the most difficult control-mode. Lastly, in between attempts, the robot would be repositioned with arbitrary rotations over the table, in order to provoke more complex operation.

Observations

Due to many similar observations of the individual participants during the usability tests, the observations will be presented in general terms, while some interesting individual cases will be described in more detail. As this is a qualitative test, the results have to be seen as informed assertions rather than definitive conclusions.

In most cases, where the participants controlled the robot well beyond the singularity warning threshold, they would immediately notice it and stop the operation. In these cases, the participants usually asked: "Is anything wrong?". A noteworthy case is one participant that had some prior knowledge of the developed system, who noted that even though he was relatively certain that he was not nearing a singular position; he instinctively stopped operation as the screen tinted. It is important to emphasize that none of the remaining participants had any prior knowledge of the singularity warning and its implications. Thus their response may be considered a natural response to the visual stimuli.

However, in other cases, the participants would not go very far beyond the warning threshold, causing the tint to be very subdued and almost imperceptible.

The participant's ability to proficiently use the system's functions though the button interface varied. Participants with some knowledge of robotics, would to a greater extent plan out the movements ahead and choose a control-mode appropriate for facilitating this motion. These participants would often align the end-effector appropriately to the workpiece and do the remaining movement using translation only and 1:2 scaling. Other participants, however, did not use the control modes in a manner that significantly benefitted them in completing the task. This was despite the fact that they seemed to understand how to use and interact with the button interface.

Regarding the participant's general level of proficiency in using the system, it was observed that many of the participants had problems understanding the perspective of control. This may be because participants related their control perspective to the camera's perspective, where the main camera would look straight towards the robot as in the screenshot previously presented in figure 6.5. This often caused a behavior where trail-and-error was used in the initial stages of a given movement, until the participant understood the connection between his and the resulting movement. One of the participants did not use the gripper analog as an actual analog to the end-effector. More specifically, with every initialization of control, the gripper analog was pitched so that it was horizontal with the gripper's fingers pointing forward, instead of mimicking the end-effectors downwards pitch. Despite this, the participant displayed good control of the robot and performed the task successfully. In general terms, all participants significantly increased their efficiency in completing the task between the two attempts. However, two of the participants displayed a level of control that the candidate would consider near full proficiency seen in the context of the systems limitations.

The following propositions are made based on the observations:

- The visual overlay turning red for communicating an approaching singularity can be an efficient way of communicating hazardous events to an operator.
- The user function interface was intuitive to understand.
- The differences in perspective for control and the camera-views' perspective, can confuse the operator.
- Efficient operation of the system can be achieved with minimal training.

Chapter 9

Discussion

The discussions will be divided into the following main topics:

- Control and programming
- Multimodal communication

Control and programming
In the thesis, an end-effector analog is used as a position-based control input device for remote operation of an industrial robot. It utilizes the combination of a Microsoft Kinect and a MARG-sensor for determining the pose of the control device. To the author's knowledge, this combination has not previously been used in this context.

The electronics associated with the MARG-sensor have a small footprint, and the Kinect is flexible by being able to quickly change the tracked object. Consequently, the combination can allow for easy interchangeability of end-effector analogs for system integrators that offers remote support to enterprises with different setups.

Despite the big differences in the method of control developed in the thesis and a traditional teach-pendant, some similarities in function can be seen. The ability to selectively mask the input to specific axis using a selection matrix, offers some similar benefits to the highly defined directional pads of a teach-pendant.

However, the use of position-based control devices may offer significant benefits in remote operation of industrial robots compared to rate-based control input devices. It can reduce the use of the traditional "move-and-wait" strategy in time-delayed situations, as the operator's kinesthetic sensations during operation make it easier to relate his/hers actions to the robot's response. Thus, larger and more complex motions can be executed in a natural way, and may result in decreased reliance on constant verification from cameras in the local room under time-delay. In addition, as presented in section 5.1, the use of an end-effector analog as a tangible interface for the control, may provide the operator with a greater understanding of how his/her actions translate to movements of the TCP, as well as some insights into the robot's required change in joint configuration.

With regard to programming, the teach-in approach can be simple to use and understand. However, it can be time-consuming if the desired path is curved. In these cases, the operator has to approximate the path with straight line-segments. This can be a tedious process with many programmed points for achieving the necessary density of points along the path. There are several approaches that can be explored for addressing this issue, which will be further elaborated on in section 10.2

All participants in the usability tests (section 8.3) successfully completed the predefined pick-and-place task. However, observations during the tests suggested that some participants would occasionally confuse the camera views' perspective for the perspective of control. There are arguably many approaches for addressing this issue. One possible approach, could be to align the control and camera's perspective by transformation of the control input. The challenges associated with this approach could be to determine which camera view is the one being used by the operator, and whether or not the camera perspective in question causes confusion for the operator. Another approach could be to mirror the camera images that cause confusion. This approach would however not address potential confusion concerning control motions that are intended to follow along the optical axis[1] of the camera in question.

Multimodal communication

The applicability of a concept for conveying information through a visual overlay, covering the operator's computer monitors was investigated as a part of the usability study in section 8.3. The example warned the operator of an approaching singularity through gradually tinting the visual overlay red. During the tests, it was observed that the behavior of the visual overlay gave rise to an immediate response of halting operation for most participants. The concept was therefore considered a viable option for conveying this information to a remote operator. However, there are many parameters and behaviors that could be used for manipulating a visual overlay. Some suggestions are:

- Intensity
- Flashing and flashing frequency
- Gradients
- Color

The overlay could also be applied to specific zones of the display. For instance, the zones could correspond to directional information such as the direction to a potential collision in the local room. Alternatively, it can be used for highlighting information of interest in parts of the operator's interface.

If the visual overlay is used uniformly over all the operator's displays, it shares some similarities with auditory feedback, in the sense that it communicates information to the operator regardless of what portion of the displays the operator is focusing on. Hence, the informa-

---

[1] Axis that points into the image (depth direction)

tion is hard to ignore which can be especially useful for conveying information that addresses safety concerns.

Moreover, in the context of safety and hazardous situations, the synthesis of both audio and visual feedback could draw even more attention as well as providing more information to the operator. This reduces the dangers associated with situations where the volume of the auditory feedback is low or turned off.

In expanding the remote operation system to provide all information for performing the major part of the end customer support of industrial robot systems, some considerations have to be taken into account. In section 4.3, it was found that transfer of video streams amounted to nearly 90% of the total bandwidth requirements for the examined hypothetical system. Hence, in expanding the system, it is important to bridge the information to an appropriate representation as compact as possible before sending it through the transfer channel.

Chapter 10

Conclusions and Recommendations for Further Work

10.1 Conclusions

Recent years have shown a trend of increased use of industrial robots in small and medium sized enterprises (SMEs). This trend presents new challenges for system integrators in how to provide instant support and knowledge-transfer to the SMEs. Remote operation of the industrial robot systems is one approach for facing these challenges. However, this further presents the challenge of how to efficiently communicate the state of the robot system to the remote operator. CogInfoCom (Cognitive Infocommunications), is a multidisciplinary field which investigates the link between the research areas of the cognitive sciences and info communications. In this thesis, a remote operation system utilizing CogInfoCom has been developed for enabling system integrators to provide instant support from their office.

The remote operation system was implemented on a 7-axis NACHI MR20 industrial robot. The system's main contents and characteristics can be summarized by the following points:

- End-effector analog used as a tangible interface for natural control and teach-in programming
- Functions for scaling, self-motion and motion guidance.
- Auditory and visual feedback for multimodal communication between the robot system and the remote operator

A series of practical experiments was performed to verify the various system functions of the remote operation system. The experiments demonstrated the following functions:

- Scaling
- Self-motion
- Motion guidance
- Robot programming

The results were that the functions successfully met their requirements.

In addition, the following was tested:

- Time-delay of one second

The observations during the test indicated that the control method aided the operator in time-delayed control, because it allowed motions to be planned ahead by relating them to operator's environment.

For testing the general usability of the remote operation system, usability tests with five participants were done. In the tests, a pick-and-place task was to be performed by the participants. All participants successfully completed the task. The observations during the tests, suggested that:

- The user interface for the various system functions was intuitive to understand.
- The differences in perspective for the camera-views and the perspective of control could confuse the participants.
- A concept for conveying information through a visual overlay, covering the operator's monitor, was an efficient way of communicating an approaching singularity.
- Efficient operation of the system can be achieved with minimal training.

The system shows that an end-effector analog as a tangible interface can provide intuitive control of an industrial robot. Moreover, it shows that the use of CogInfoCom and multimodal man-machine communication, provide successful and efficient communication with a remote operator controlling an industrial robot.

10.2 Recommendations for Further Work

The thesis holds several elements that could be further investigated and expanded upon. Four main topics are recommended for further work:

- Programming of line segments.
- Exploring the use of a visual overlay and it's implications for communication.
- Control perspective in remote operation systems with multi-camera setups.

The use of a position-based control input device, opens for many approaches for programming a robot. For addressing the issues regarding tedious programming of curved paths, one possible approach would be to continuously save the points or joint angles during programming. However, this approach puts high requirements on the operator's ability to control precisely while recording the given path. Arguably, a more intriguing approach could be to use the gripper analog to define start and end points of a B-spline, and subsequently using it for defining intermediate control points of the desired path. Also, displaying the path in a 3D-model of the work space could give the operator the ability to easily verify the path.

The possible behaviors of the visual overlay and their implications for communication of information to a remote operator is a topic of interest. A qualitative study focusing on how its parameters affects a remote operator, and what type of information is appropriate could be performed. The results could yield a clearer picture of its possibilities and limitations.

How to best solve the issues of confusing the control and camera perspectives in remote operation systems utilizing multi-camera setups, is considered an important topic for further work. If a operator confuses his/hers perspective in a highly time-delayed situation and executes a planned motion, the use of a position-based control device could prove to be a potential hazard rather than a benefit in time-delayed operation. Further studies into this issue could therefore yield a control method that is intuitive for more people as well as addressing the associated safety concerns.

Additional research in remote operation of industrial robots has previously been done at PPM's laboratories in Trondheim. In light of this, effort could be made in further unifying the important aspects of all research for creating a remote operation system that addresses all aspects of performing the major part of end customer support remotely.

Bibliography

ABB.com (2012). ABB.com.

Aracil, R., Buss, M., Cobos, S., Ferre, M., Hirche, S., Kuschel, M., and Peer, A. (2007). The Human Role in Telerobotics. *Springer Tracts in Advanced Robotics*, 31:11–24.

Ascension-tech.com (2012). Ascension-tech.com.

Atkinson, R. L., Atkinson, R. C., Smith, E. E., Bem, D. J., and Nolen-Hoeksema, S. (1999). *Hilgards's Introduction to Psychology.* Harcourt College Publishers, 13th edition.

Baranyi, P. and Csapo, A. (2010). Cognitive infocommunications: Coginfocom. In *Computational Intelligence and Informatics (CINTI), 2010 11th International Symposium on*, pages 141–146. IEEE.

Baranyi, P., Persa, G., and Csapo, A. (2011). Definition of Cognitive Infocommunications and an Architectural Implementation of Cognitive Infocommunications Systems. *World Academy of Science, Engineering and Technology*, (58):501–505.

Basanez, L. and Suarez, R. (2009). Teleoperation. In *Springer Handbook of Automation*, pages 449–468. Springer.

Bejczy, A. (2001). Teleoperation and telerobotics. In Nwokah, O. D. . I. . and Hurmuzlu, Y., editors, *The Mechanical Systems Design Handbook.* CRC Press.

Bejczy, A. and Handlykken, M. (1981). Generalization of Bilateral Force-Reflecting Control of Manipulators. In *4th CISM Symp. on Theory and Practice of Robots and Manipulators*, Poland.

Biggs, G. and Macdonald, B. (2003). A Survey of Robot Programming Systems. In *Proceedings of the Australasian Conference on Robotics and Automation.*

Brooks, T. and Bejczy, A. (1985). Hand Controllers for Teleoperation: A State-of-the-Art Technology Survey and Evaluation. *JPL Publication 85-11.*

Burdea, G. C. and Member, S. (1999). Invited Review : The Synergy Between Virtual Reality and Robotics. 15(3):400–410.

Buss, M., Carton, D., and Gonsior, B. (2011). Towards proactive human-robot interaction in human environments. In *Cognitive Infocommunications (CogInfoCom), 2011 2nd International Conference on Cognitive Infocommunications*, pages 1–6.

CogInfoCom.hu (2012). http://coginfocom.hu/.

Csapo, A. and Baranyi, P. (2011). A unified terminology for CogInfoCom applications. In *Cognitive Infocommunications (CogInfoCom), 2011 2nd International Conference on Cognitive Infocommunications*, pages 1–6. IEEE.

Fanucrobotics.com (2012). www.fanucrobotics.com.

Gagné, J. (2011). Intuitive Teleoperation of an Intelligent Robotic System Using Low Cost 6 DOF Motion Capture. *Master Thesis, University of Waterloo.*

Galambos, P. and Baranyi, P. (2011). Vibrotactile force feedback for telemanipulation: Concept and applications. In *Cognitive Infocommunications (CogInfoCom), 2011 2nd International Conference on Cognitive Infocommunications*, pages 1–6. IEEE.

Gonzalez, R., Fu, K., and Lee, C. (1987). *Robotics: Control, Sensing, Vision, and Intelligence.* Mcgraw-Hill Book Company.

Han, J., Dalton, J., Vaughan, B., Oertel, C., Dougherty, C., De Looze, C., and Campbell, N. (2011). Collecting multi-modal data of human-robot interaction. In *Cognitive Infocommunications (CogInfoCom), 2011 2nd International Conference on Cognitive Infocommunications*, pages 1–4.

Hendseth, S. (1994). *Model-Based Control in Telemanipulation.* PhD thesis, NTH, Norway.

Horne, R. (1978). MANTIS - A compact mobile remote handling system for accelerator halls and tunnels. Technical report, CERN, Geneva, Switzerland.

Horne, R., Lohmann, K., Coull, L., Coin, A., Therville, A., Lips, R., Desrozier, M., and Hunt, K. (1987). MANTIS 2 - A new long range remote vehicle and servo-master-slave manipulator for the cern accelerator complex. Technical report, CERN, Geneva, Switzerland.

Horne, R. A. (1988). Teleoperator Evolution at CERN. Technical report, CERN, Geneva, Switzerland.

HUNOROB (2012). http://www.itm-lab.org/academic-research-and-r-d-programs/hunorob-program.

JARA (2012). http://www.jara.jp/e/.

Karwowski, W. and Rahimi, M. (1992). *Human-Robot Interaction.* Taylor & Francis.

Khoshelham, K. and Elberink, S. O. (2012). Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors*, 12(2):1437–1454.

Kuka-robotics.com (2012). Kuka-robotics.com.

Lepetit, V. and Fua, P. (2005). Monocular Model-Based 3D Tracking of Rigid Objects : A Survey. *Foundations and Trendsin Computer Graphics and Vision*, 1(1):1–89.

Madgwick, S. O. H., Harrison, A. J. L., and Vaidyanathan, A. (2011). Estimation of IMU and MARG orientation using a gradient descent algorithm. *IEEE International Conference on Rehabilitation Robotics*, 2011:5975346.

Moeslund, T. B. and Granum, E. (2001). A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding*, 81(3):231–268.

Motoman.com (2012). Motoman.com.

Nachirobotics.com (2012). Nachirobotics.com.

Niemeyer, G., Preusche, C., and Hirzinger, G. (2008). Telerobotics. In *Springer Handbook of Robotics*, pages 741–757.

Nof, S. Y., editor (1999). *Handbook of Industrial Robotics*. Wiley, New York, 2nd editio edition.

Paperno, E. and Keisar, P. (2004). Three-Dimensional Magnetic Tracking of Biaxial Sensors. 40(3):1530–1536.

Paperno, E., Sasada, I., and Leonovich, E. (2001). A New Method for Magnetic Position and Orientation Tracking. *IEEE TRANSACTIONS ON MAGNETICS*, 37(4):1938–1940.

Petercorke.com (2012). http://www.petercorke.com/Robotics_Toolbox.html.

PrimeSense (2012). http://www.primesense.com/en/technology.

Raab, F. h., Blood, E. B., Steiner, T. O., and Jones, H. R. (1979). Magnetic Position and Orientation Tracking System. *IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS*, 15(5):709–718.

Rasmussen, N. T. r., St, M., Moeslund, T. B., and Granum, E. (2001). REAL-TIME TRACKING FOR VIRTUAL ENVIRONMENTS USING SCAAT KALMAN FILTERING AND UNSYNCHRONISED CAMERAS.

Reme, F. and Sanderud, A. R. (2011). Sensor based control of industrial robots. *Specialization Project, Norwegian University of Science and Technology*.

Rolland, J. P., Baillot, Y., and Goon, A. (2001). A SURVEY OF TRACKING TECHNOLOGY FOR VIRTUAL ENVIRONMENTS. *Center for Research and Education in Optics and Lasers*, University:1–48.

Romer.eu (2012). Romer.eu.

SMErobot (2012). http://www.smerobot.org/ - The European Robot Initiative for Strengthening the Competitiveness of SMEs in Manufacturing.

Sokolov, E. N. (1962). HIGHER NERVOUS FUNCTIONS: THE ORIENTING REFLEX.

Thomessen, P. D. T. and Lien, T. K. (2004). Intuitive robot programming. In *ISR 35th International Symposium on Robotics*, Paris, France.

Thomessen, T. and Kosicki, T. (2011). Cognitive audio-visual infocommunication applied in remote support for industrial robot systems. In *Cognitive Infocommunications (CogInfoCom), 2011 2nd International Conference on Cognitive Infocommunications*, pages 1–5. IEEE.

Thomessen, T., Nagy, L., and Olah, K. (2011). Remote control of industrial robot systems using cognitive info communication. In *Cognitive Infocommunications (CogInfoCom), 2011 2nd International Conference on Cognitive Infocommunications*, pages 1–6. IEEE.

VIRCA-Website (2012). http://www.virca.hu/.

ViSP (2012). http://www.irisa.fr/lagadic/visp/visp.html.

Whitney, D. (1985). Historical perspective and state of the art in robot force control. In *IEEE International Conference on Robotics and Automation*.

List of Figures

List of Tables

Appendix A

Acronyms

CROP Cognitive Remote Operation

DH Denavit-Hartenberg Formulation

DOF Degrees of Freedom

RGB Red Green Blue

ROI Remote Operator Interface

RTC Robotic Technology Component

RTM Robotics Technology Middleware

UDP User Datagram Protocol

VI Virtual Instrument

Appendix B

Rotation matrices

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{bmatrix} \tag{B.1}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} \tag{B.2}$$

$$\mathbf{R}_z(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.3}$$

Appendix C

Contents of digital attachment

The digital attachment contains:

- Remote operation interface (Processing programming environment)
- Remote operation server (LabVIEW)
- Gripper Analog Firmware for Arduino microcontroller.
- Videos documenting the experiments, see table C.1 for list of videos

C.1 Video reference table

Video #	File name	Description	Related section
Video 1	video1-scaling	Scaling experiment	Section 8.1.1
Video 2	video2-selfmotion	Self motion experiment	Section 8.1.2
Video 3	video3-motionguidance	Motion guidance experiment	Section 8.1.3
Video 4	video4-programming	Robot programming experiment	Section 8.1.4
Video 5	video5-timedelay	Time delay experiment	Section 8.2

Appendix D

Robot Program

The following program was the result of the robot programming experiment in section 8.1.4. All values prefixed by & are hexadecimals for servo encoder values.

```
MOVEX A=7,AC=0,SM=0,M1E,L,(&H07FA8DE, &H0800000, &H07FD149, &H07FFF43,
&H08067E4, &H080014D),S=300,H=1,MS,M2E,P,(&H07FFFB0),S=300,H=1
MOVEX A=7,AC=0,SM=0,M1E,L,(&H07FBB0F, &H0800000, &H07FF8E2, &H07FFEF7,
&H0808A80, &H0800393),S=300,H=1,MS,M2E,P,(&H0800387),S=300,H=1
MOVEX A=7,AC=0,SM=0,M1E,L,(&H07FADF7, &H0800000, &H08011C3, &H0800427,
&H0809269, &H07FE51C),S=300,H=1,MS,M2E,P,(&H07FD393),S=300,H=1
MOVEX A=7,AC=0,SM=0,M1E,L,(&H07FA0AA, &H0800000, &H07FE058, &H080065F,
&H0806CD9, &H07FDCBC),S=300,H=1,MS,M2E,P,(&H07FC90C),S=300,H=1
MOVEX A=7,AC=0,SM=0,M1E,L,(&H07FAA0B, &H0800000, &H08008C9, &H08005B8,
&H0808AED, &H07FDC61),S=300,H=1,MS,M2E,P,(&H07FC649),S=300,H=1
MOVEX A=7,AC=0,SM=0,M1E,L,(&H07F9B3F, &H0800000, &H080202A, &H080868C,
&H08087C0, &H07FE105),S=300,H=1,MS,M2E,P,(&H08026BA),S=300,H=1
MOVEX A=7,AC=0,SM=0,M1E,L,(&H07F89E7, &H0800000, &H0803A10, &H0800A60,
&H0806758, &H0800341),S=300,H=1,MS,M2E,P,(&H0800106),S=300,H=1
```

Appendix E

Participants

Overview of participants:

#	Age	Gender	Occupation
1	23	F	MSc in biotechnology
2	25	M	MSc in production systems
3	26	M	PhDc in the field of robotics
4	18	M	High-school student
5	18	M	High-school student

Appendix F

Libraries and Code

Razor AHRS
Razor AHRS is an open source firmware for making the Arduino in conjunction with the 9DOF Sensor Stick to work as an Attitude and Heading Reference system.

It is maintained by Peter Bartz and Sacha Spors, and can be found at:

https://dev.qu.tu-berlin.de/projects/sf-razor-9dof-ahrs

Simple OpenNI
Simple OpenNI is an OpenNI and NITE wrapper for Processing. It is used for the Kinect tracking, and can be found at:

http://code.google.com/p/simple-openni/

Hypermedia UDP library
Used for the UDP sockets in the Remote Operation Interface, and can be found at:

http://ubaa.net/shared/processing/udp/

IPCapture library
The IPCapture library is used for displaying the MJPEG streams from the web cameras, and can be found at:

http://code.google.com/p/ipcapture/

LabView Robotics module
Used for forward and inverse kinematic manipulations on the LabView server, and can be found at:

http://www.ni.com/labview/robotics/

Appendix G

Calibration of AHRS System

The sensors has to be calibrated to establish the baseline gravity readings for all accelerometers, drift in the gyros and the baseline magnetic readings for all axis. The following calibration was established and used in the thesis:

```
// Accelerometer
// "accel x,y,z (min/max) = X_MIN/X_MAX  Y_MIN/Y_MAX  Z_MIN/Z_MAX"
#define ACCEL_X_MIN ((float) -260)
#define ACCEL_X_MAX ((float) 260)
#define ACCEL_Y_MIN ((float) -252)
#define ACCEL_Y_MAX ((float) 270)
#define ACCEL_Z_MIN ((float) -257)
#define ACCEL_Z_MAX ((float) 262)

// Magnetometer
// "magn x,y,z (min/max) = X_MIN/X_MAX  Y_MIN/Y_MAX  Z_MIN/Z_MAX"
#define MAGN_X_MIN ((float) -650)
#define MAGN_X_MAX ((float) 347)
#define MAGN_Y_MIN ((float) -422)
#define MAGN_Y_MAX ((float) 600)
#define MAGN_Z_MIN ((float) -558)
#define MAGN_Z_MAX ((float) 475)

// Gyroscope
// "gyro x,y,z (current/average) = .../OFFSET_X  .../OFFSET_Y  .../OFFSET_Z
#define GYRO_AVERAGE_OFFSET_X ((float) 9.3)
#define GYRO_AVERAGE_OFFSET_Y ((float) 18.9)
#define GYRO_AVERAGE_OFFSET_Z ((float) -10.2)
```

Appendix H

Datasheets

Datasheets for the following is included:

- NACHI MR20 Industrial Robot
- Schunk PG-70
- Olimex SAM9-L9260
- 9DOF Sensor Stick Digital Compass - HMC5883L
- 9DOF Sensor Stick Digital Accelerometer - ADXL345
- 9DOF Sensor Stick Digital Gyroscope - ITG-3200
- Arduino Uno Microcontroller - ATmega328
- Cisco WVC210 Wireless-G (PTZ) Internet Video Camera

Due to the lack of flyer-versions of the datasheets for most of these, the included documents are excerpts that is deemed to give a brief introduction to their specifications. There is no general datasheet for the 9DOF Sensor Stick, Arduino Uno and Microsoft Kinect. Therefore, for the 9DOF Sensor Stick, excerpts from each of it's sensors is included. For the Arduino Uno, which is based on the Atmel ATmega328 microcontroller, an excerpt for the ATmega328 8-bit AVR microcontroller datasheet is included. Lastly, for the Microsoft Kinect, there is no official technical datasheet. However, a overview over it's components and links to their datasheets can be found at:

`http://openkinect.org/wiki/Hardware_info#Datasheets`

H.1 NACHI MR20

Presto

プレスト MR20/20L

NACHI

動き自在、7軸“腕”ロボット
Flexible motion “Arm” robot with 7-axes

プレスト MR20/20L

より複雑な動作を可能にする7軸構造を採用

- これまでの6軸構造では実現できなかった狭いスペースや障害物のある場所でのロボット適用が可能に!

コンパクトボディ・パワフルアーム

- 省スペースレイアウトが可能。コンパクトなボディに余裕の可搬質量20kg　最大30kg(*1)

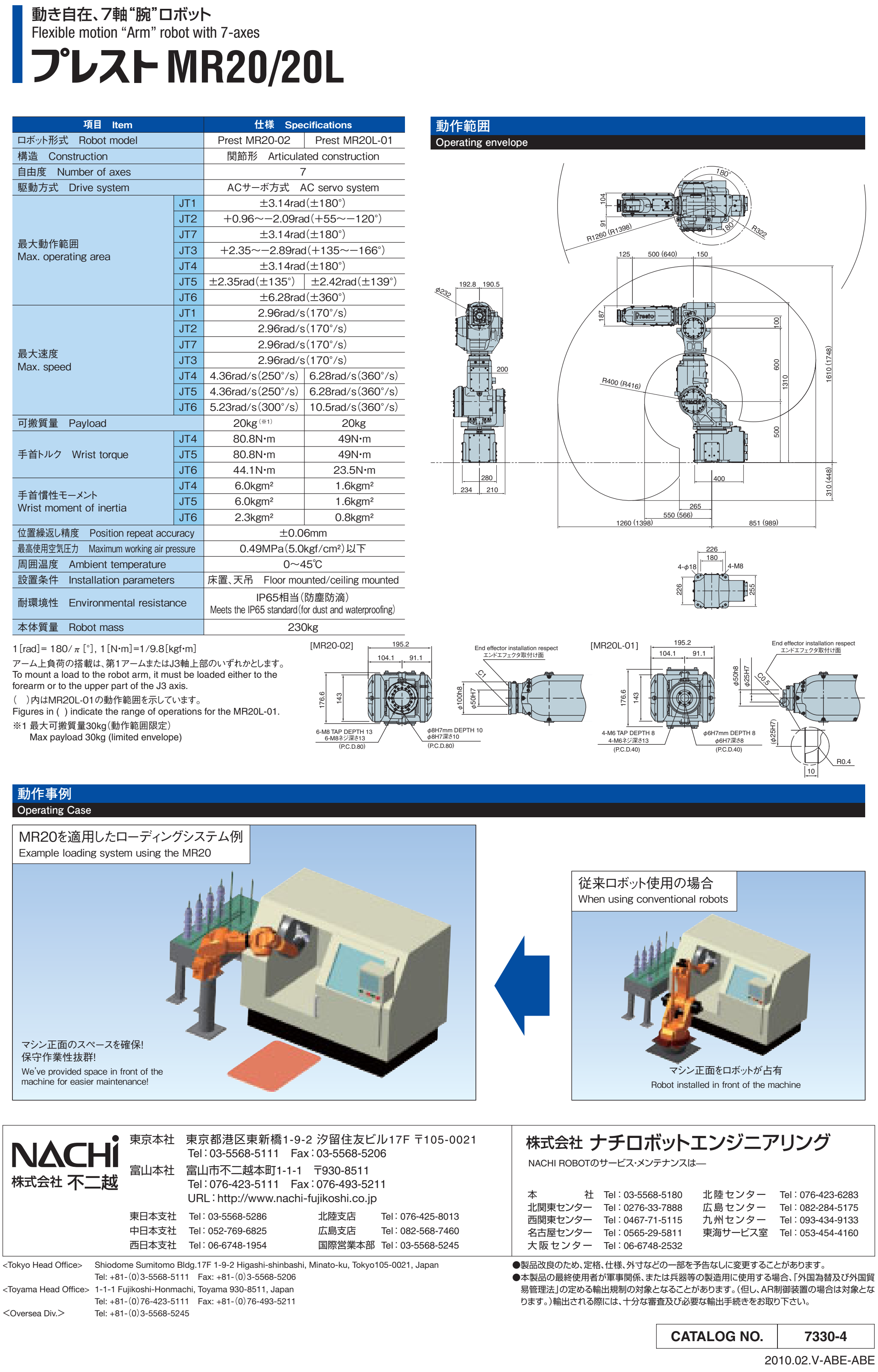
7-axes structure

- Flexible and complex positioning and motion can be available by 7-axes structure.

Compact body, powerful arm

- Minimizing installation space.
- Payload 20kg MAX 30kg(*1)

(*1) 可搬質量30kg時は動作範囲に制限があります。
Limited envelope within 30kg



動き自在、7軸“腕”ロボット
Flexible motion “Arm” robot with 7-axes

プレスト MR20/20L

項目　Item		仕様　Specifications	
ロボット形式　Robot model		Prest MR20-02	Prest MR20L-01
構造　Construction		関節形　Articulated construction	
自由度　Number of axes		7	
駆動方式　Drive system		ACサーボ方式　AC servo system	
最大動作範囲 Max. operating area	JT1	±3.14rad(±180°)	
	JT2	+0.96〜−2.09rad(+55〜−120°)	
	JT7	±3.14rad(±180°)	
	JT3	+2.35〜−2.89rad(+135〜−166°)	
	JT4	±3.14rad(±180°)	
	JT5	±2.35rad(±135°)	±2.42rad(±139°)
	JT6	±6.28rad(±360°)	
最大速度 Max. speed	JT1	2.96rad/s(170°/s)	
	JT2	2.96rad/s(170°/s)	
	JT7	2.96rad/s(170°/s)	
	JT3	2.96rad/s(170°/s)	
	JT4	4.36rad/s(250°/s)	6.28rad/s(360°/s)
	JT5	4.36rad/s(250°/s)	6.28rad/s(360°/s)
	JT6	5.23rad/s(300°/s)	10.5rad/s(360°/s)
可搬質量　Payload		20kg(※1)	20kg
手首トルク　Wrist torque	JT4	80.8N·m	49N·m
	JT5	80.8N·m	49N·m
	JT6	44.1N·m	23.5N·m
手首慣性モーメント Wrist moment of inertia	JT4	6.0kgm²	1.6kgm²
	JT5	6.0kgm²	1.6kgm²
	JT6	2.3kgm²	0.8kgm²
位置繰返し精度　Position repeat accuracy		±0.06mm	
最高使用空気圧力　Maximum working air pressure		0.49MPa(5.0kgf/cm²)以下	
周囲温度　Ambient temperature		0〜45℃	
設置条件　Installation parameters		床置、天吊　Floor mounted/ceiling mounted	
耐環境性　Environmental resistance		IP65相当(防塵防滴) Meets the IP65 standard(for dust and waterproofing)	
本体質量　Robot mass		230kg	

1[rad]= 180/π[°], 1[N·m]=1/9.8[kgf·m]

アーム上負荷の搭載は、第1アームまたはJ3軸上部のいずれかとします。
To mount a load to the robot arm, it must be loaded either to the forearm or to the upper part of the J3 axis.

(　)内はMR20L-01の動作範囲を示しています。
Figures in ( ) indicate the range of operations for the MR20L-01.

※1 最大可搬質量30kg(動作範囲限定)
Max payload 30kg (limited envelope)

動作範囲
Operating envelope

動作事例
Operating Case

MR20を適用したローディングシステム例
Example loading system using the MR20

マシン正面のスペースを確保!
保守作業性抜群!
We've provided space in front of the machine for easier maintenance!

従来ロボット使用の場合
When using conventional robots

マシン正面をロボットが占有
Robot installed in front of the machine

NACHI
株式会社 不二越

東京本社　東京都港区東新橋1-9-2 汐留住友ビル17F 〒105-0021
Tel：03-5568-5111　Fax：03-5568-5206
富山本社　富山市不二越本町1-1-1　〒930-8511
Tel：076-423-5111　Fax：076-493-5211
URL：http://www.nachi-fujikoshi.co.jp

東日本支社	Tel：03-5568-5286	北陸支店	Tel：076-425-8013
中日本支社	Tel：052-769-6825	広島支店	Tel：082-568-7460
西日本支社	Tel：06-6748-1954	国際営業本部	Tel：03-5568-5245

株式会社 ナチロボットエンジニアリング
NACHI ROBOTのサービス・メンテナンスは—

本　社	Tel：03-5568-5180	北陸センター	Tel：076-423-6283
北関東センター	Tel：0276-33-7888	広島センター	Tel：082-284-5175
西関東センター	Tel：0467-71-5115	九州センター	Tel：093-434-9133
名古屋センター	Tel：0565-29-5811	東海サービス室	Tel：053-454-4160
大阪センター	Tel：06-6748-2532		

<Tokyo Head Office>　Shiodome Sumitomo Bldg.17F 1-9-2 Higashi-shinbashi, Minato-ku, Tokyo105-0021, Japan
Tel: +81-(0)3-5568-5111　Fax: +81-(0)3-5568-5206
<Toyama Head Office>　1-1-1 Fujikoshi-Honmachi, Toyama 930-8511, Japan
Tel: +81-(0)76-423-5111　Fax: +81-(0)76-493-5211
<Oversea Div.>　Tel: +81-(0)3-5568-5245

- ●製品改良のため、定格、仕様、外寸などの一部を予告なしに変更することがあります。
- ●本製品の最終使用者が軍事関係、または兵器等の製造用に使用する場合、「外国為替及び外国貿易管理法」の定める輸出規制の対象となることがあります。(但し、AR制御装置の場合は対象となります。)輸出される際には、十分な審査及び必要な輸出手続きをお取り下さい。

CATALOG NO.　7330-4

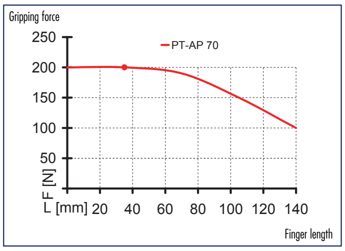
2010.02.V-ABE-ABE

H.2 Schunk PG-70

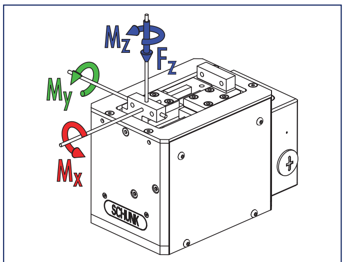
PG 70

Electrical · 2-Finger Parallel Gripper · Universal Gripper

Gripping force, I.D. gripping



Finger load



M_x max. 17.6 Nm
M_y max. 17.6 Nm
M_z max. 18.4 Nm
F_z max. 200.0 N

ⓘ Moments and forces apply per base jaw and may occur simultaneously. M_y may arise in addition to the moment generated by the gripping force itself. If the max. permitted finger weight is exceeded, it is imperative to throttle the air pressure so that the jaw movement occurs without any hitting or bouncing. Service life may be reduced.

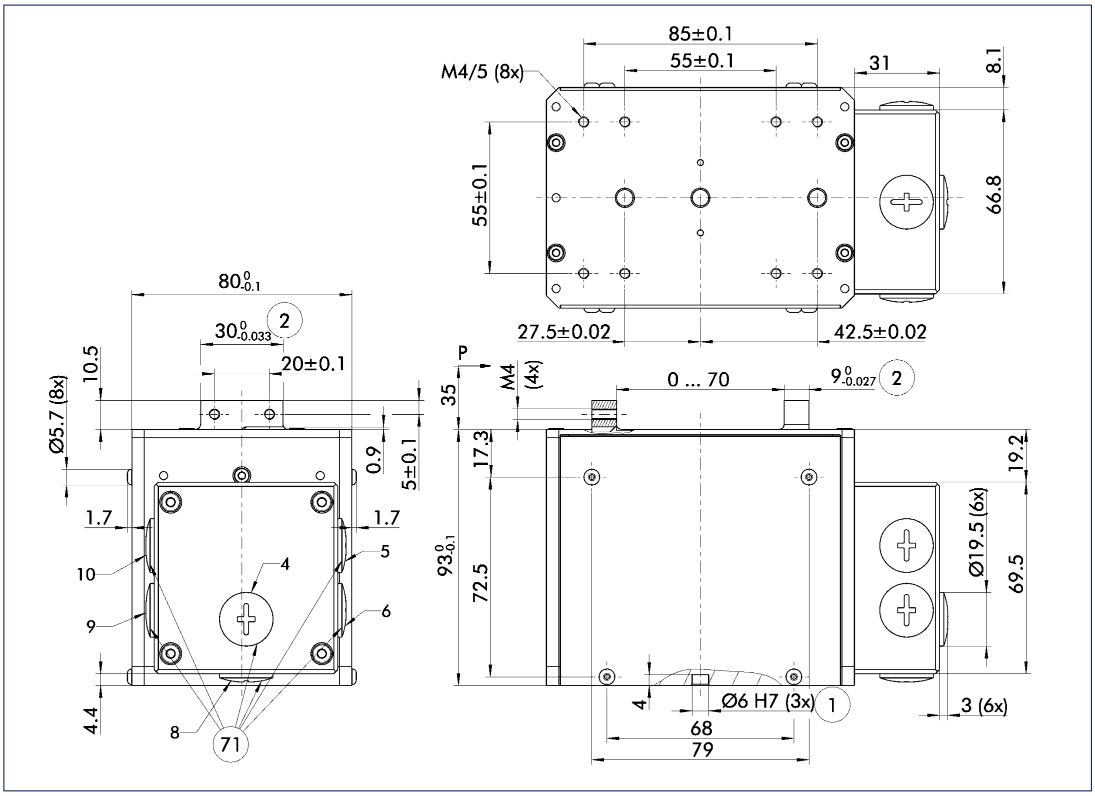
Technical data

Description		PG 70
Mechanical gripper operating data	ID	0306090
Stroke per finger	[mm]	35.0
Constant gripping force (100 % continuous duty)	[N]	200.0
Max. gripping force	[N]	200.0
Min. gripping force	[N]	30.0
Weight	[kg]	1.4
Recommended workpiece weight	[kg]	1.0
Closing time	[s]	1.1
Opening time	[s]	1.1
Max. permitted finger length	[mm]	140.0
IP class		20
Min. ambient temperature	[°C]	5.0
Max. ambient temperature	[°C]	55.0
Repeat accuracy	[mm]	0.05
Positioning accuracy	[mm]	on request
Max. velocity	[mm/s]	82.0
Max. acceleration	[mm/s²]	328.0
Electrical operating data for gripper		
Terminal voltage	[V]	24.0
Nominal power current	[A]	1.8
Maximum current	[A]	6.5
Resolution	[μm]	1.0
Controller operating data		
Integrated electronics		Yes
Voltage supply	[VDC]	24.0
Nominal power current	[A]	0.5
Sensor system		Encoder
Interface		I/O, RS 232, CAN-Bus, Profibus DP

PG 70

Electrical · 2-Finger Parallel Gripper · Universal Gripper

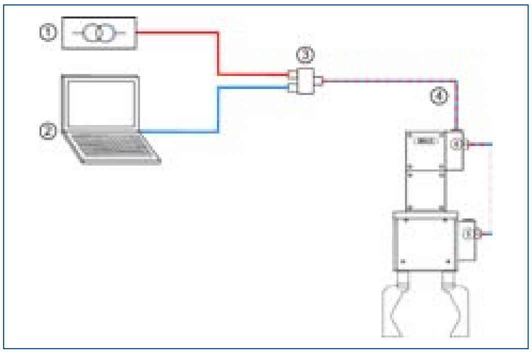
Main views



The drawing shows the gripper in the basic version with closed jaws, the dimensions do not include the options described below.

① Gripper connection
② Finger connection
(71) M16x1.5 for cable gland

Actuation



① 24 VDC power supply provided by the customer
② Control (PLC or similar) provided by the customer
③ PAE 130 TB terminal block (ID No. 0307725) for connecting the power supply, the communication and the hybrid cable
④ Hybrid cable for connecting the PowerCube modules

Interconnecting cable

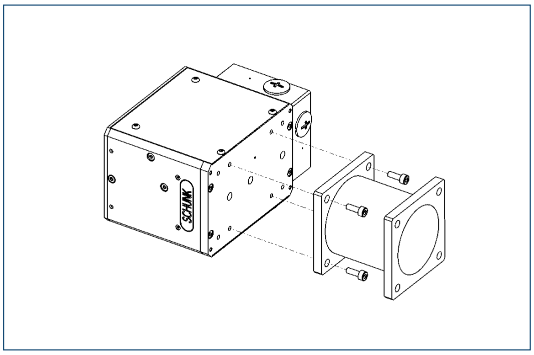
Description	ID	Length
PowerCube Hybrid cable, coiled	0307753	0.3 m
PowerCube Hybrid cable, coiled	0307754	0.5 m
PowerCube Hybrid cable, straight (per meter)	9941120	

ⓘ The 'Hybrid cable' is recommended for the use in CAN-Bus- or RS232-systems.
For Profibus applications we recommend to use a separate standardized Profibus cable for the communication.
You can find further cables in the „Accessories" catalog section.

PG 70

Electrical · 2-Finger Parallel Gripper · Universal Gripper

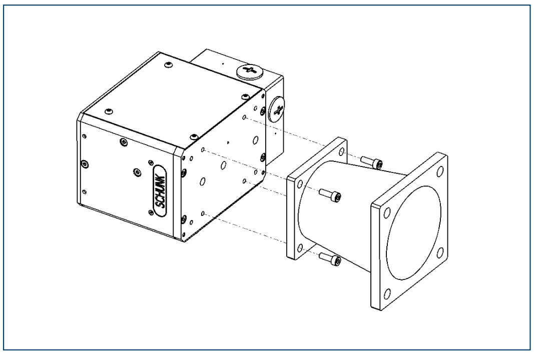
Mechanical accessories



Straight connecting elements

Description	ID	Dimensions
PAM 100	**0307800**	70x70/35/70x70 mm
PAM 101	**0307801**	70x70/70/70x70 mm

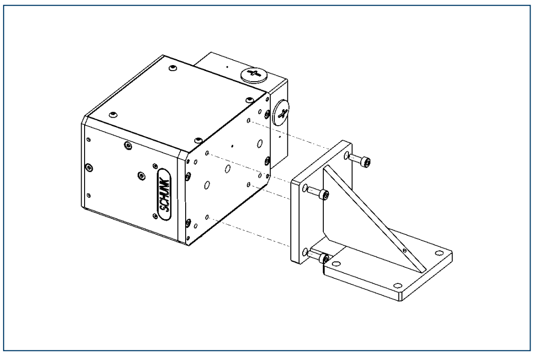
Special lengths on request
Straight standard element for connecting size 70 PowerCube modules



Conical connecting elements

Description	ID	Dimensions
PAM 110	**0307810**	90x90/45/70x70 mm
PAM 111	**0307811**	90x90/90/70x70 mm

Special lengths on request
Conical standard element for connecting size 70 and 90 PowerCube modules



Right-angle connecting elements

Description	ID	Dimensions
PAM 120	**0307820**	90°/70.5x98

Special lengths on request
Right-angle standard element for connecting size 70 PowerCube modules

You can find more detailed information and individual parts of the above-mentioned accessories in the „Accessories“ catalog section.

H.3 Olimex SAM9-L9260

INTRODUCTION:

SAM9-L9260 is a low cost development platform with ARM9 microcontroller, 64MB SDRAM and 512MB NAND Flash. The board has Ethernet 100Mbit controller, USB host, USB device, RS232 and 40 pin extension port with all unused SAM9260 ports available for add-on boards. SAM9-L9260 has waste amount of Flash and RAM and runs Linux, WindowsCE and other RTOS natively. The on-board RTC clock is equipped with a 3V Li backup battery.

BOARD FEATURES:

- MCU: AT91SAM9260 16/32 bit ARM9™ 200MHz operation
- 50MHz system (main) clock
- standard JTAG connector with ARM 2x10 pin layout for programming/debugging with ARM-JTAG
- 64 MB SDRAM
- 512MB NAND Flash (seen in Linux as silicon drive)
- Ethernet 100Mbit connector
- USB host and USB device connectors
- RS232 interface and drivers
- SD/MMC card connector
- one user button and one reset button
- one power and two status LEDs
- on board voltage regulator 3.3V with up to 800mA current
- single power supply: 5V DC required
- power supply filtering capacitor
- 18.432 Mhz crystal on socket
- extension header
- PCB: FR-4, 1.5 mm (0,062"), soldermask, silkscreen component print
- Dimensions: 100 x 80 mm (3.94 x 3.15")

ELECTROSTATIC WARNING:

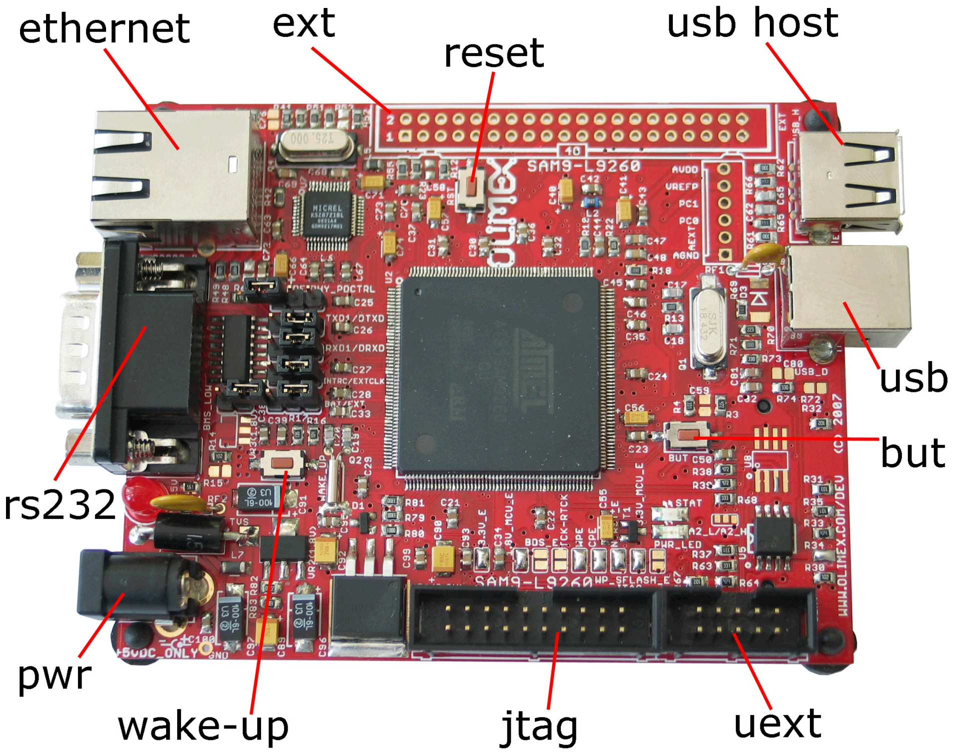
The SAM9-L9260 board is shipped in protective anti-static packaging. The board must not be subject to high electrostatic potentials. General practice for working with static sensitive devices should be applied when working with this board.

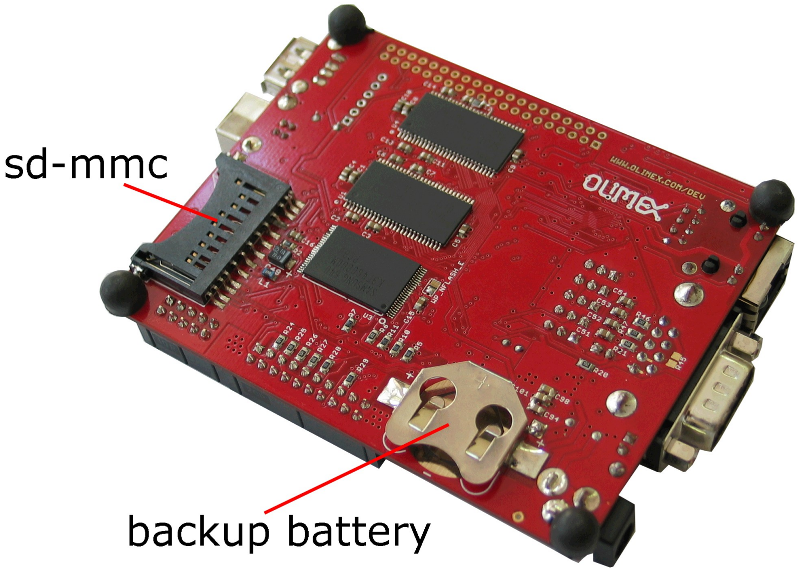
BOARD USE REQUIREMENTS:

Cables: 1.8 meter USB A-B cable to connect with USB host.
Null modem RS232 female – female to connect with PC COM port.
Hardware: **ARM-JTAG, ARM-USB-OCD** or other compatible tool if you want to program this board with JTAG, usually with linux installed you can develop without the need for JTAG.

Software: The CD contains Linux 2.6 complete with source and binary in CD.

BOARD LAYOUT:





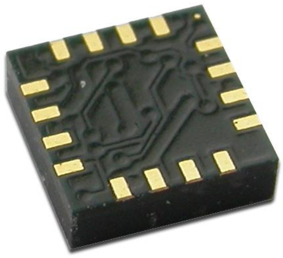
H.4 Olimex 9DOF Sensor Stick Digital Compass - HMC5883L

Three-Axis Digital Compass IC HMC5883L

Honeywell

Advanced Information

The Honeywell HMC5883L is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as low-cost compassing and magnetometry. The HMC5883L includes our state-of-the-art, high-resolution HMC118X series magneto-resistive sensors plus an ASIC containing amplification, automatic degaussing strap drivers, offset cancellation, and a 12-bit ADC that enables 1° to 2° compass heading accuracy. The I^2C serial bus allows for easy interface. The HMC5883L is a 3.0x3.0x0.9mm surface mount 16-pin leadless chip carrier (LCC). Applications for the HMC5883L include Mobile Phones, Netbooks, Consumer Electronics, Auto Navigation Systems, and Personal Navigation Devices.



The HMC5883L utilizes Honeywell's Anisotropic Magnetoresistive (AMR) technology that provides advantages over other magnetic sensor technologies. These anisotropic, directional sensors feature precision in-axis sensitivity and linearity. These sensors' solid-state construction with very low cross-axis sensitivity is designed to measure both the direction and the magnitude of Earth's magnetic fields, from milli-gauss to 8 gauss. Honeywell's Magnetic Sensors are among the most sensitive and reliable low-field sensors in the industry.

FEATURES	BENEFITS
Three-Axis Magnetoresistive Sensors and ASIC in a 3.0x3.0x0.9mm LCC Surface Mount Package	Small Size for Highly Integrated Products. Just Add a Micro-Controller Interface, Plus Two External SMT Capacitors Designed for High Volume, Cost Sensitive OEM Designs Easy to Assemble & Compatible with High Speed SMT Assembly
12-Bit ADC Coupled with Low Noise AMR Sensors Achieves 5 milli-gauss Resolution in ±8 Gauss Fields	Enables 1° to 2° Degree Compass Heading Accuracy
Built-In Self Test	Enables Low-Cost Functionality Test after Assembly in Production
Low Voltage Operations (2.16 to 3.6V) and Low Power Consumption (100 μA)	Compatible for Battery Powered Applications
Built-In Strap Drive Circuits	Set/Reset and Offset Strap Drivers for Degaussing, Self Test, and Offset Compensation
I^2C Digital Interface	Popular Two-Wire Serial Data Interface for Consumer Electronics
Lead Free Package Construction	RoHS Compliance
Wide Magnetic Field Range (+/-8 Oe)	Sensors Can Be Used in Strong Magnetic Field Environments with a 1° to 2° Degree Compass Heading Accuracy
Software and Algorithm Support Available	Compassing Heading, Hard Iron, Soft Iron, and Auto Calibration Libraries Available
Fast 160 Hz Maximum Output Rate	Enables Pedestrian Navigation and LBS Applications

HMC5883L

SPECIFICATIONS (* Tested and specified at 25°C except stated otherwise.)

Characteristics	Conditions*	Min	Typ	Max	Units
Power Supply					
Supply Voltage	VDD Referenced to AGND	2.16		3.6	Volts
	VDDIO Referenced to DGND	1.71	1.8	VDD+0.1	Volts
Average Current Draw	Idle Mode	-	2	6	μA
	Measurement Mode (7.5 Hz ODR; No measurement average, MA1:MA0 = 00)	-	100	-	μA
	Specified at: VDD = 2.5V, VDDIO = 1.8V				
Performance					
Field Range	Full scale (FS) – total applied field (Typical)	-8		+8	gauss
Mag Dynamic Range	3-bit gain control	±1		±8	gauss
Resolution	VDD=3.0V, GN=2		5		milli-gauss
Linearity	±2.0 gauss input range			0.1	±% FS
Hysteresis	±2.0 gauss input range		±25		ppm
Cross-Axis Sensitivity	Test Conditions: Cross field = 0.5 gauss, Happlied = ±3 gauss		±0.2%		%FS/gauss
Output Rate (ODR)	Continuous Measurment Mode	0.75		75	Hz
	Single Measurement Mode			160	Hz
Measurement Period	From receiving command to data ready		6		msec
Turn-on Time	Ready for I2C commands		200		μs
Gain Tolerance	All gain/dynamic range settings		±5		%
I^{2}C Address	7-bit address		0x1E		hex
	8-bit read address		0x3D		hex
	8-bit write address		0x3C		hex
I^{2}C Rate	Controlled by I^{2}C Master			400	kHz
I^{2}C Hysteresis	Hysteresis of Schmitt trigger inputs on SCL and SDA - Fall (VDDIO=1.8V)		0.2*VDDIO		Volts
	Rise (VDDIO=1.8V)		0.8*VDDIO		Volts
Self Test	X & Y Axes		±1.16		gauss
	Z Axis		±1.08		
	X & Y Axes (GN=100)		440		LSb
	Z Axis (GN=100)				
General					
ESD Voltage	Human Body Model (all pins)			2000	Volts
	CDM			750	
Operating Temperature	Ambient	-30		85	°C
Storage Temperature	Ambient, unbiased	-40		125	°C
Reflow Classification	MSL 3, 260 °C Peak Temperature				
Package Size	Length and Width	2.85	3.00	3.15	mm
Package Height		0.8	0.9	1.0	mm
Package Weight			18		mg

HMC5883L

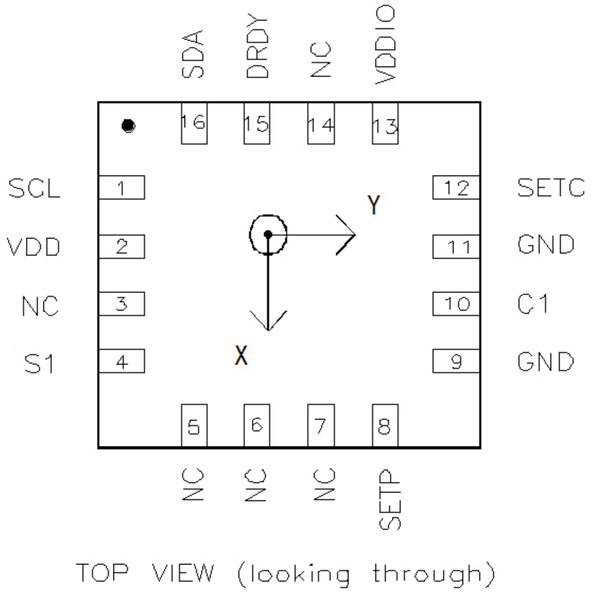
Absolute Maximum Ratings (* Tested at 25°C except stated otherwise.)

Characteristics	Min	Max	Units
Supply Voltage VDD	-0.3	4.8	Volts
Supply Voltage VDDIO	-0.3	4.8	Volts

PIN CONFIGURATIONS

Pin	Name	Description
1	SCL	Serial Clock – I^{2}C Master/Slave Clock
2	VDD	Power Supply (2.16V to 3.6V)
3	NC	Not to be Connected
4	S1	Tie to VDDIO
5	NC	Not to be Connected
6	NC	Not to be Connected
7	NC	Not to be Connected
8	SETP	Set/Reset Strap Positive – S/R Capacitor (C2) Connection
9	GND	Supply Ground
10	C1	Reservoir Capacitor (C1) Connection
11	GND	Supply Ground
12	SETC	S/R Capacitor (C2) Connection – Driver Side
13	VDDIO	IO Power Supply (1.71V to VDD)
14	NC	Not to be Connected
15	DRDY	Data Ready, Interrupt Pin. Internally pulled high. Optional connection. Low for 250 μsec when data is placed in the data output registers.
16	SDA	Serial Data – I^{2}C Master/Slave Data

Table 1: Pin Configurations



H.5 9DOF Sensor Stick Digital Accelerometer - ADXL345

ANALOG
DEVICES

3-Axis, ±2 *g*/±4 *g*/±8 *g*/±16 *g* Digital Accelerometer

ADXL345

FEATURES

- **Ultralow power: as low as 23 µA in measurement mode and 0.1 µA in standby mode at V_S = 2.5 V (typical)**
- **Power consumption scales automatically with bandwidth**
- **User-selectable resolution**
  - **Fixed 10-bit resolution**
  - **Full resolution, where resolution increases with *g* range, up to 13-bit resolution at ±16 *g* (maintaining 4 m*g*/LSB scale factor in all *g* ranges)**
- **Patent pending, embedded memory management system with FIFO technology minimizes host processor load**
- **Single tap/double tap detection**
- **Activity/inactivity monitoring**
- **Free-fall detection**
- **Supply voltage range: 2.0 V to 3.6 V**
- **I/O voltage range: 1.7 V to V_S**
- **SPI (3- and 4-wire) and I^2C digital interfaces**
- **Flexible interrupt modes mappable to either interrupt pin**
- **Measurement ranges selectable via serial command**
- **Bandwidth selectable via serial command**
- **Wide temperature range (−40°C to +85°C)**
- **10,000 *g* shock survival**
- **Pb free/RoHS compliant**
- **Small and thin: 3 mm × 5 mm × 1 mm LGA package**

APPLICATIONS

- **Handsets**
- **Medical instrumentation**
- **Gaming and pointing devices**
- **Industrial instrumentation**
- **Personal navigation devices**
- **Hard disk drive (HDD) protection**

GENERAL DESCRIPTION

The ADXL345 is a small, thin, ultralow power, 3-axis accelerometer with high resolution (13-bit) measurement at up to ±16 *g*. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I^2C digital interface.

The ADXL345 is well suited for mobile device applications. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (3.9 m*g*/LSB) enables measurement of inclination changes less than 1.0°.

Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion by comparing the acceleration on any axis with user-set thresholds. Tap sensing detects single and double taps in any direction. Free-fall sensing detects if the device is falling. These functions can be mapped individually to either of two interrupt output pins. An integrated, patent pending memory management system with a 32-level first in, first out (FIFO) buffer can be used to store data to minimize host processor activity and lower overall system power consumption.

Low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation.

The ADXL345 is supplied in a small, thin, 3 mm × 5 mm × 1 mm, 14-lead, plastic package.

FUNCTIONAL BLOCK DIAGRAM

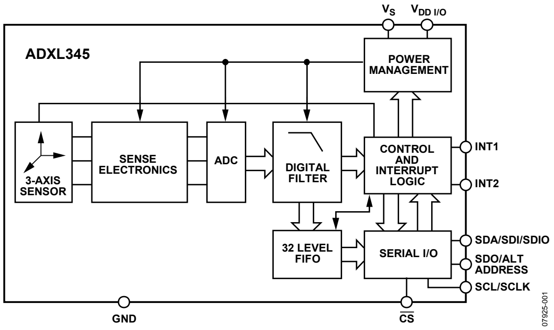


Figure 1.

Rev. C

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113

SPECIFICATIONS

$T_A = 25°C$, $V_S = 2.5$ V, $V_{DD\ I/O} = 1.8$ V, acceleration = 0 *g*, $C_S = 10$ µF tantalum, $C_{I/O} = 0.1$ µF, output data rate (ODR) = 800 Hz, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Test Conditions	Min	Typ[1]	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range	User selectable		±2, ±4, ±8, ±16		*g*
Nonlinearity	Percentage of full scale		±0.5		%
Inter-Axis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity[2]			±1		%
OUTPUT RESOLUTION	Each axis				
All *g* Ranges	10-bit resolution		10		Bits
±2 *g* Range	Full resolution		10		Bits
±4 *g* Range	Full resolution		11		Bits
±8 *g* Range	Full resolution		12		Bits
±16 *g* Range	Full resolution		13		Bits
SENSITIVITY	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT}	All *g*-ranges, full resolution	230	256	282	LSB/*g*
	±2 *g*, 10-bit resolution	230	256	282	LSB/*g*
	±4 *g*, 10-bit resolution	115	128	141	LSB/*g*
	±8 *g*, 10-bit resolution	57	64	71	LSB/*g*
	±16 *g*, 10-bit resolution	29	32	35	LSB/*g*
Sensitivity Deviation from Ideal	All *g*-ranges		±1.0		%
Scale Factor at X_{OUT}, Y_{OUT}, Z_{OUT}	All *g*-ranges, full resolution	3.5	3.9	4.3	m*g*/LSB
	±2 *g*, 10-bit resolution	3.5	3.9	4.3	m*g*/LSB
	±4 *g*, 10-bit resolution	7.1	7.8	8.7	m*g*/LSB
	±8 *g*, 10-bit resolution	14.1	15.6	17.5	m*g*/LSB
	±16 *g*, 10-bit resolution	28.6	31.2	34.5	m*g*/LSB
Sensitivity Change Due to Temperature			±0.01		%/°C
0 *g* OFFSET	Each axis				
0 *g* Output for X_{OUT}, Y_{OUT}		−150	0	+150	m*g*
0 *g* Output for Z_{OUT}		−250	0	+250	m*g*
0 *g* Output Deviation from Ideal, X_{OUT}, Y_{OUT}			±35		m*g*
0 *g* Output Deviation from Ideal, Z_{OUT}			±40		m*g*
0 *g* Offset vs. Temperature for X-, Y-Axes			±0.4		m*g*/°C
0 *g* Offset vs. Temperature for Z-Axis			±1.2		m*g*/°C
NOISE					
X-, Y-Axes	ODR = 100 Hz for ±2 *g*, 10-bit resolution or all *g*-ranges, full resolution		0.75		LSB rms
Z-Axis	ODR = 100 Hz for ±2 *g*, 10-bit resolution or all *g*-ranges, full resolution		1.1		LSB rms
OUTPUT DATA RATE AND BANDWIDTH	User selectable				
Output Data Rate (ODR)[3, 4, 5]		0.1		3200	Hz
SELF-TEST[6]					
Output Change in X-Axis		0.20		2.10	*g*
Output Change in Y-Axis		−2.10		−0.20	*g*
Output Change in Z-Axis		0.30		3.40	*g*
POWER SUPPLY					
Operating Voltage Range (V_S)		2.0	2.5	3.6	V
Interface Voltage Range ($V_{DD\ I/O}$)		1.7	1.8	V_S	V
Supply Current	ODR ≥ 100 Hz		140		µA
	ODR < 10 Hz		30		µA
Standby Mode Leakage Current			0.1		µA
Turn-On and Wake-Up Time[7]	ODR = 3200 Hz		1.4		ms

Parameter	Test Conditions	Min	Typ[1]	Max	Unit
TEMPERATURE					
Operating Temperature Range		−40		+85	°C
WEIGHT					
Device Weight			30		m*g*

[1] The typical specifications shown are for at least 68% of the population of parts and are based on the worst case of mean ±1 σ, except for 0 *g* output and sensitivity, which represents the target value. For 0 *g* offset and sensitivity, the deviation from the ideal describes the worst case of mean ±1 σ.

[2] Cross-axis sensitivity is defined as coupling between any two axes.

[3] Bandwidth is the −3 dB frequency and is half the output data rate, bandwidth = ODR/2.

[4] The output format for the 3200 Hz and 1600 Hz ODRs is different than the output format for the remaining ODRs. This difference is described in the Data Formatting of Upper Data Rates section.

[5] Output data rates below 6.25 Hz exhibit additional offset shift with increased temperature, depending on selected output data rate. Refer to the Offset Performance at Lowest Data Rates section for details.

[6] Self-test change is defined as the output (*g*) when the SELF_TEST bit = 1 (in the DATA_FORMAT register, Address 0x31) minus the output (*g*) when the SELF_TEST bit = 0. Due to device filtering, the output reaches its final value after 4 × τ when enabling or disabling self-test, where τ = 1/(data rate). The part must be in normal power operation (LOW_POWER bit = 0 in the BW_RATE register, Address 0x2C) for self-test to operate correctly.

[7] Turn-on and wake-up times are determined by the user-defined bandwidth. At a 100 Hz data rate, the turn-on and wake-up times are each approximately 11.1 ms. For other data rates, the turn-on and wake-up times are each approximately τ + 1.1 in milliseconds, where τ = 1/(data rate).

H.6 9DOF Sensor Stick Digital Gyroscope - ITG-3200

InvenSense	**ITG-3200 Product Specification**	Document Number: PS-ITG-3200A-00-01.4 Revision: 1.4 Release Date: 03/30/2010

1.2 Purpose and Scope

This document is a preliminary product specification, providing a description, specifications, and design related information for the ITG-3200^{TM}. Electrical characteristics are based upon simulation results and limited characterization data of advanced samples only. Specifications are subject to change without notice. Final specifications will be updated based upon characterization of final silicon.

1.3 Product Overview

The ITG-3200 is the world's first single-chip, digital-output, 3-axis MEMS gyro IC optimized for gaming, 3D mice, and 3D remote control applications. The part features enhanced bias and sensitivity temperature stability, reducing the need for user calibration. Low frequency noise is lower than previous generation devices, simplifying application development and making for more-responsive remote controls.

The ITG-3200 features three 16-bit analog-to-digital converters (ADCs) for digitizing the gyro outputs, a user-selectable internal low-pass filter bandwidth, and a Fast-Mode I^{2}C (400kHz) interface. Additional features include an embedded temperature sensor and a 2% accurate internal oscillator. This breakthrough in gyroscope technology provides a dramatic 67% package size reduction, delivers a 50% power reduction, and has inherent cost advantages compared to competing multi-chip gyro solutions.

By leveraging its patented and volume-proven Nasiri-Fabrication platform, which integrates MEMS wafers with companion CMOS electronics through wafer-level bonding, InvenSense has driven the ITG-3200 package size down to a revolutionary footprint of 4x4x0.9mm (QFN), while providing the highest performance, lowest noise, and the lowest cost semiconductor packaging required for handheld consumer electronic devices. The part features a robust 10,000*g* shock tolerance, as required by portable consumer equipment.

For power supply flexibility, the ITG-3200 has a separate VLOGIC reference pin, in addition to its analog supply pin, VDD, which sets the logic levels of its I^{2}C interface. The VLOGIC voltage may be anywhere from 1.71V min to VDD max.

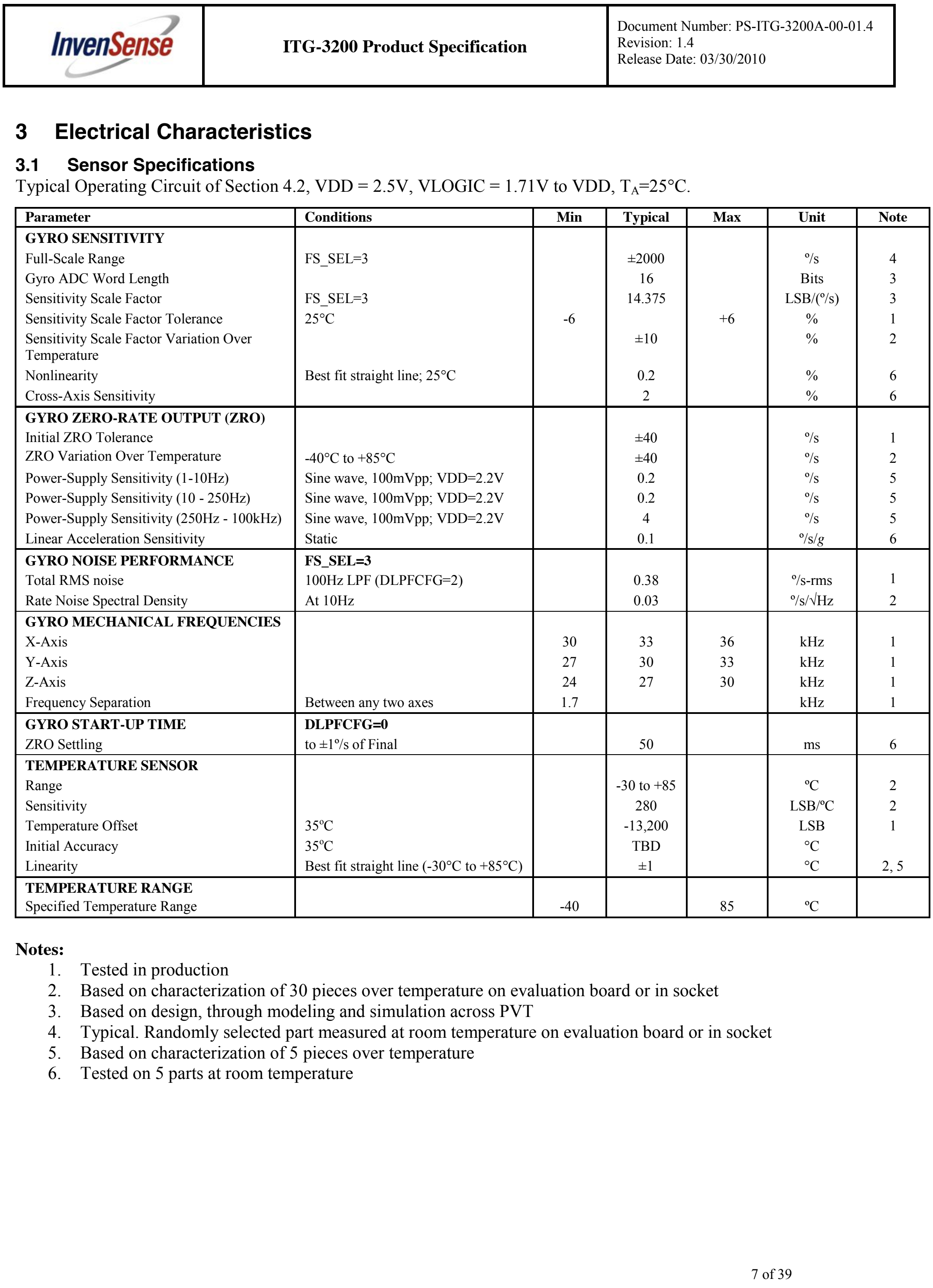
1.4 Applications

- Motion-enabled game controllers
- Motion-based portable gaming
- Motion-based 3D mice and 3D remote controls
- "No Touch" UI
- Health and sports monitoring

2 Features

The ITG-3200 triple-axis MEMS gyroscope includes a wide range of features:

- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyros) on one integrated circuit with a sensitivity of 14.375 LSBs per °/sec and a full-scale range of ±2000°/sec
- Three integrated 16-bit ADCs provide simultaneous sampling of gyros while requiring no external multiplexer
- Enhanced bias and sensitivity temperature stability reduces the need for user calibration
- Low frequency noise lower than previous generation devices, simplifying application development and making for more-responsive motion processing
- Digitally-programmable low-pass filter
- Low 6.5mA operating current consumption for long battery life
- Wide VDD supply voltage range of 2.1V to 3.6V
- Flexible VLOGIC reference voltage allows for I^{2}C interface voltages from 1.71V to VDD
- Standby current: 5µA
- Smallest and thinnest package for portable devices (4x4x0.9mm QFN)
- No high pass filter needed
- Turn on time: 50ms
- Digital-output temperature sensor
- Factory calibrated scale factor
- 10,000 *g* shock tolerant
- Fast Mode I^{2}C (400kHz) serial interface
- On-chip timing generator clock frequency is accurate to +/-2% over full temperature range
- Optional external clock inputs of 32.768kHz or 19.2MHz to synchronize with system clock
- MEMS structure hermetically sealed and bonded at wafer level
- RoHS and Green compliant



3 Electrical Characteristics

3.1 Sensor Specifications

Typical Operating Circuit of Section 4.2, VDD = 2.5V, VLOGIC = 1.71V to VDD, T_A=25°C.

Parameter	Conditions	Min	Typical	Max	Unit	Note
GYRO SENSITIVITY						
Full-Scale Range	FS_SEL=3		±2000		°/s	4
Gyro ADC Word Length			16		Bits	3
Sensitivity Scale Factor	FS_SEL=3		14.375		LSB/(°/s)	3
Sensitivity Scale Factor Tolerance	25°C	-6		+6	%	1
Sensitivity Scale Factor Variation Over Temperature			±10		%	2
Nonlinearity	Best fit straight line; 25°C		0.2		%	6
Cross-Axis Sensitivity			2		%	6
GYRO ZERO-RATE OUTPUT (ZRO)						
Initial ZRO Tolerance			±40		°/s	1
ZRO Variation Over Temperature	-40°C to +85°C		±40		°/s	2
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp; VDD=2.2V		0.2		°/s	5
Power-Supply Sensitivity (10 - 250Hz)	Sine wave, 100mVpp; VDD=2.2V		0.2		°/s	5
Power-Supply Sensitivity (250Hz - 100kHz)	Sine wave, 100mVpp; VDD=2.2V		4		°/s	5
Linear Acceleration Sensitivity	Static		0.1		°/s/*g*	6
GYRO NOISE PERFORMANCE	**FS_SEL=3**					
Total RMS noise	100Hz LPF (DLPFCFG=2)		0.38		°/s-rms	1
Rate Noise Spectral Density	At 10Hz		0.03		°/s/√Hz	2
GYRO MECHANICAL FREQUENCIES						
X-Axis		30	33	36	kHz	1
Y-Axis		27	30	33	kHz	1
Z-Axis		24	27	30	kHz	1
Frequency Separation	Between any two axes	1.7			kHz	1
GYRO START-UP TIME	**DLPFCFG=0**					
ZRO Settling	to ±1°/s of Final		50		ms	6
TEMPERATURE SENSOR						
Range			-30 to +85		ºC	2
Sensitivity			280		LSB/ºC	2
Temperature Offset	35ºC		-13,200		LSB	1
Initial Accuracy	35ºC		TBD		°C	
Linearity	Best fit straight line (-30°C to +85°C)		±1		°C	2, 5
TEMPERATURE RANGE						
Specified Temperature Range		-40		85	ºC	

Notes:

1. Tested in production
2. Based on characterization of 30 pieces over temperature on evaluation board or in socket
3. Based on design, through modeling and simulation across PVT
4. Typical. Randomly selected part measured at room temperature on evaluation board or in socket
5. Based on characterization of 5 pieces over temperature
6. Tested on 5 parts at room temperature

H.7 Arduino Uno Microcontroller - ATmega328

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash progam memory (ATmega48PA/88PA/168PA/328P)
  - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
  - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
    Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I^2C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
  - Active Mode: 0.2 mA
  - Power-down Mode: 0.1 μA
  - Power-save Mode: 0.75 μA (Including 32 kHz RTC)

8-bit AVR® Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash

ATmega48PA
ATmega88PA
ATmega168PA
ATmega328P

Rev. 8161D–AVR–10/09

H.8 Cisco WVC210 Wireless-G (PTZ) Internet Video Camera

Data Sheet

Cisco WVC210 Wireless-G Pan Tilt Zoom (PTZ) Internet Video Camera: 2-Way Audio
Cisco Small Business Video Surveillance Cameras

High-Quality, Flexible, Remote-Controlled Wireless Video Solution for Your Small Business

Highlights

- High-quality remote-controlled wireless video camera
- Captures images even in low-light environments (1 lux at f2.0)
- Simultaneous dual codecs provide optimal combination of viewing and storage of video
- Supports two-way audio, IP multicast, 3GPP (3rd Generation Partnership Project), and more advanced features

Figure 1. Cisco WVC210 Wireless-G PTZ Internet Video Camera: 2-Way Audio

Product Overview

Cisco® Small Business Video Surveillance products provide customizable ways for small business owners to monitor and protect their companies. These high-quality solutions can be optimized for many different applications and sites.

The Cisco WVC210 Wireless-G PTZ Internet Video Camera (Figure 1) sends live video through the Internet to a web browser anywhere in the world. The camera supports dual codecs (MPEG-4 and MJPEG), which can be used simultaneously. MPEG-4 gives efficient bandwidth consumption with good-quality compression and is optimal for real-time viewing of video. MJPEG gives optimal video quality, making it ideal for large-volume storage to a network attached storage (NAS) device.

The Cisco WVC210's audio capabilities include two-way audio, an embedded microphone, external speaker and microphone ports, and voice compression. With extensive support for features such as IP multicast, Real Time Streaming Protocol (RTSP), Real Time Protocol (RTP), and 3rd Generation Partnership Project (3GPP), it enables video to be viewed from multiple endpoints and client applications, such as 3G phones and QuickTime clients on PCs or Wi-Fi phones. Network protocols such as 802.1p priority, 802.1Q VLANs, and Dynamic DNS (DDNS) are also supported.

The pan/tilt and digital zoom functions allow you to remotely control the camera movement and focus, giving you maximum remote flexibility. Up to 10 simultaneous unicast users can access the camera at any time. Video monitoring software is included for monitoring multiple cameras and recording to your hard drive, with advanced search by time and date. Recording can be set up to start by motion trigger or by manual or scheduled recording. Playback is available on Windows Media Player, with no need for a proprietary player.

You can also enable security mode, which tells the camera to send a message with a short attached video to up to three email addresses whenever it detects motion in its field of view. You can then log in to the live video stream if the situation warrants. Wireless security features include Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), and WPA2.

Features

- Pan, tilt, and 2x digital zoom
- Complementary metal-oxide semiconductor (CMOS) sensor with glass lens
- Dual codecs (MPEG-4 and MJPEG) supported simultaneously
- Captures video and two-way audio (with built-in microphone and external speaker) to your hard drive
- Built-in web server for remote access over IP
- Supports Universal Plug and Play (UPnP) for easy discovery on the network
- DDNS support for available free DDNS services
- Supports up to 10 simultaneous unicast users
- Motion detection with event notification to an email account or alarm log in the monitoring software
- LCD screen displays full IP address for easy configuration
- Includes software for monitoring, recording, and playback of up to 16 cameras
- Captures JPEG snapshots at multiple resolutions; snapshots can be sent to an FTP server
- IP multicast - supports unlimited multicast users
- Real Time Streaming Protocol (RTSP) video and audio streaming to unicast and multicast clients
- 3GPP allows viewing of video on a 3G mobile device
- Real-Time video recording from the Web interface directly with one button recording
- PC-less event recording directly to Network Attached Storage via integrated Samba client

Appendix I

Pre-study Report

NTNU – Trondheim
Norwegian University of
Science and Technology

Pre-study report

Advanced Remote Control of Industrial Robots

Stud.techn. Fredrik Reme

Spring 2012

Department of Production and Quality Engineering
Norwegian University of Science and Technology
in Partial Fulfillment of the requirements
for the Degree of Master of Science

Supervisor: Professor II Trygve Thomessen

Preface

This is a pre-study report for the master thesis (course code: TPK4900), which is in partial fulfillment of the requirements for the degree of Master of Science. It is a part of an integrated masters degree in mechanical engineering at NTNU, Trondheim. The master thesis is rewarded 30 credits, which is equivalent to a workload of approx. 48 hours pr week. The thesis will be evaluated on the basis of a written report, as well as other material pertaining to it, which will be submitted to the Department of Production and Quality Engineering (IPK) within June 11th 2012.

The thesis will be carried out by stud.techn Fredrik Reme on behalf of PPM AS and NTNU, and will contain both theoretical studies as well as practical implementations.

The problem description has been developed by Prof. Trygve Thomessen and concerns advanced remote control of industrial robots. The problem is aimed at small and medium sized companies (SMEs), where there are limited access to in-house experts to provide the necessary knowledge, training and close support for use of industrial robots to be successfull. The goal of the thesis is to develop new methods and technology for enabling the system integrators to do the major part of the end customer support remotely. A practical solution proposal will be developed in PPMs laboratory in Trondheim for a NACHI MR20 industrial robot. Through this, the candidate will learn about the subjectmatter, practical work in a research environment as well as project management.

The pre-study is performed in order to gain an overview of the thesis' tasks and scope at an early stage. The documents in the pre-study report will be used as tools for providing pointers for both time and resources needed to retain an appropriate progression in the work.

Contents

Acronyms

CogInfoCom Cognitive Infocommunications

CROP Cognitive Remote Operation

DoF Degrees of Freedom

ROS Remote Operation System

1 Problem Description

In this section, the problem to be solved in the master-thesis will be described through by presenting the background and how its ties to the various fields this master-thesis will touch on. It will also present the the components of the preliminary plan-of-action for answering the questions of this research.

1.1 Background

The increasing labour costs in the western countries the during last years has changed the landscape for traditional production companies. It has warranted a push for increased use of industrial robots in a growing set of industrial tasks. While large enterprises can afford to have in-house staff with competence and expertise within the field of robotics and production automation, this may not be the case for small and medium sized enterprises (SMEs).

The discrepancy between the SMEs need for advanced technology like robotics while having limited competence, provides new challenges for the system integrators and their end customers. The challenge revolves around how to handle the SMEs higher requirements for support and knowledge transfer to be able to sufficiently utilize an industrial robot system.

For tackling this problem, this master thesis will focus on development of new methods and technology for doing the major part of the customer support remotely by efficiently communicate the state of the physical robot cell to a remote user. The research will entail structuring and developing a laboratory setup for performing advanced remote control of industrial robots based on Cognitive Infocommunications, CogInfoCom.

The primary goal of Cognitive infocommunications according to Baranyi and Csapo (2010) is:

> Provide a complete view of how brain processes can be merged with info communications devices so that the cognitive capabilities of the human brain may not only be efficiently extended through these devices, irrespective of geographical distance, but may also be efficiently matched withe the capabilities of any artificially cognitive system

Humans are highly dependent on using all of their trained senses while performing everyday tasks. In robot operation for example, stereo- and peripheral vision is important for depth perception and spacial awareness, which make the operator able to skillfully control the robot around obstacles and singularities etc.. In traditional remote control of industrial robots, however, this is among some of the information that is lost, causing effective control to become difficult.

By including the field of cognitive info communications, the research wants to explore whether a Kinect sensor device can be used for teach-in programming and investigate if it can provide a substitute or be of assistance for the loss of spacial awareness in remote operation.

1.2 Overview and Approach

The Cognitive Remote Operation (CROP) system, was developed as a part of the HUNOROB research project. It is considered to be a relative feature complete system for remote operation of industrial robots with full 3D presentation and camera views of the physical robot cell, audio indicators on various states of the physical robot cell etc.

At PPM AS' laboratories, the CROP system has been implemented with a Nachi MR20 industrial robot by ehernet connections to the Nachi AX20 controller. Unfortunately, there are some indications that the AX20 gives low priority to commands sent over ethernet, causing unwanted delay between the remote and physical system.

Together with the supervisor, it has been agreed upon that the CROP system will be used as a framework for testing methods and technology that is to be explored during the research. Therefore, the thesis work will include restructuring of the CROP system to perform communication regarding the current internal state of the robot using the Olimex high-speed interface.

The preliminary control scheme for teach-in programming that has been chosen to be explored as a part of the thesis can be explained as follows:

> Tracking data from the Kinect and potential supporting technology will give the operator the ability to perform direct manipulation of the robots end-effector by translating and rotating either his hand or an object (e.g. an analog to the physical robots end-effector).

All practical implementations will be done in an experimental environment at PPM AS' offices using a Nachi MR20 7DoF industrial robot with AX20 controller, an Olimex high-speed interface and other supporting equipment.

By analyzing the resulting solution from the practical implementations, the research aims to answer the two main questions:

- How can such a system aid SMEs and system integrators in the challenge of knowledge transfer?
- Does the teach-in programming control scheme give additional value in remote operation of industrial robots?

Caveats regarding the delivered problem description

The supervisor is in agreement that the task no. II in the problem description, *"Introductory study of teach-in programming using wireless devices. Develop methodology and system structure for this."*, is to be considered as "Kinect" instead of "wireless devices".

2 Project Partners

NTNU

The Norwegian University of Science and Technology is the main provider of higher education in technological and natural science in Norway. Amongst the 20 000 students, more than 10 000 are studying technological subjects. The department of production and quality engineering is situated under the Faculty of Engineering Science and Technology. The department is focusing on education and research in three areas; Production Systems, Product Management and Reliability, Availability, Maintainability and Safety. The department has extensive experience with project- and master thesis in close cooperation with the industry.

PPM AS

PPM (Productive Programming Methods) was founded in December 2000 by Dr.ing Trygve Thomessen and Siv.ing Per Kristian Sannæs. PPM has it's main focus on R&D projects on productivity improvement and robotics in low batch production.

3 Project planning

This chapter will utilise various tools for project planning and control described by Rolstad (2006). This is done in order to gain an overview of the project's tasks and scope at an early stage. The following documents will be presented:

Project overview Statement
Overview over the projects problem, goals, success criteria, conditions and risks.

Work Breakdown Structure
Is a one-dimensional breakdown of the work. The project is broken down into smaller elements in a logical and systematical manner. A WBS can assist in identifying the most important parts of the project and their relationship to each other and the project as a whole.

Gantt
Is a well-known planning and scheduling tool which shows the various activities vs. time in a diagram. It provides a good visual presentation of all the activities and their duration as well as sequence.

Work packages
Is used to give a more in-depth view of what the various work packages entails. Each work package is presented in standardized tables in order to give fast and easy access to its information.

Milestones
Table of all milestones and the planned date of reaching them.

3.1 Project Overview Statement

Project Overview Statement			
Project:	Advanced Remote Control of Industrial Robots	**Responsible:**	Stud.techn. Fredrik Reme

Problem:

The increasing labor costs in the western countries the during last years has changed the landscape for traditional production companies. It has warranted a push for increased use of industrial robots in a growing set of industrial tasks. While large enterprises can afford to have in-house staff with competence and expertise within the field of robotics and production automation, this may not be the case for small and medium sized enterprises (SMEs).

Advanced remote control of industrial robots has been used as a tool for bridging the gap between the SMEs and the system integrators, enabling more effective knowledge transfer. The field of advanced remote control of industrial robots is to be explored by this project work by investigating new methods for teach-in programming as well as other technology.

Main Goal:

Develop and investigate new methods and technology for doing the major part of the customer support remotely on a NACHI MR20 by efficiently communicating the state of the physical robot cell to a remote user.

Secondary Goals

- Write a preliminary report
- Discuss how remote control can be simplified/assisted by real time sensor feedback
- Discuss how CogInfoCom can be used for efficiently communicating between a remote operator and a physical robot
- Develop a specification and structure for the information channel between the remote operation system and physical robot system.
- Develop and implement method(s) for teach-in programming using Kinect
- Develop and implement system for advanced remote operation of industrial robot based on CogInfoCom.
- Write a status report
- Write a final report.

Success Criteria:

- The produced material meet the expectations of the partners in the project
- Get top grade
- The report is adaptable to a paper that can be published

Conditions and risks:

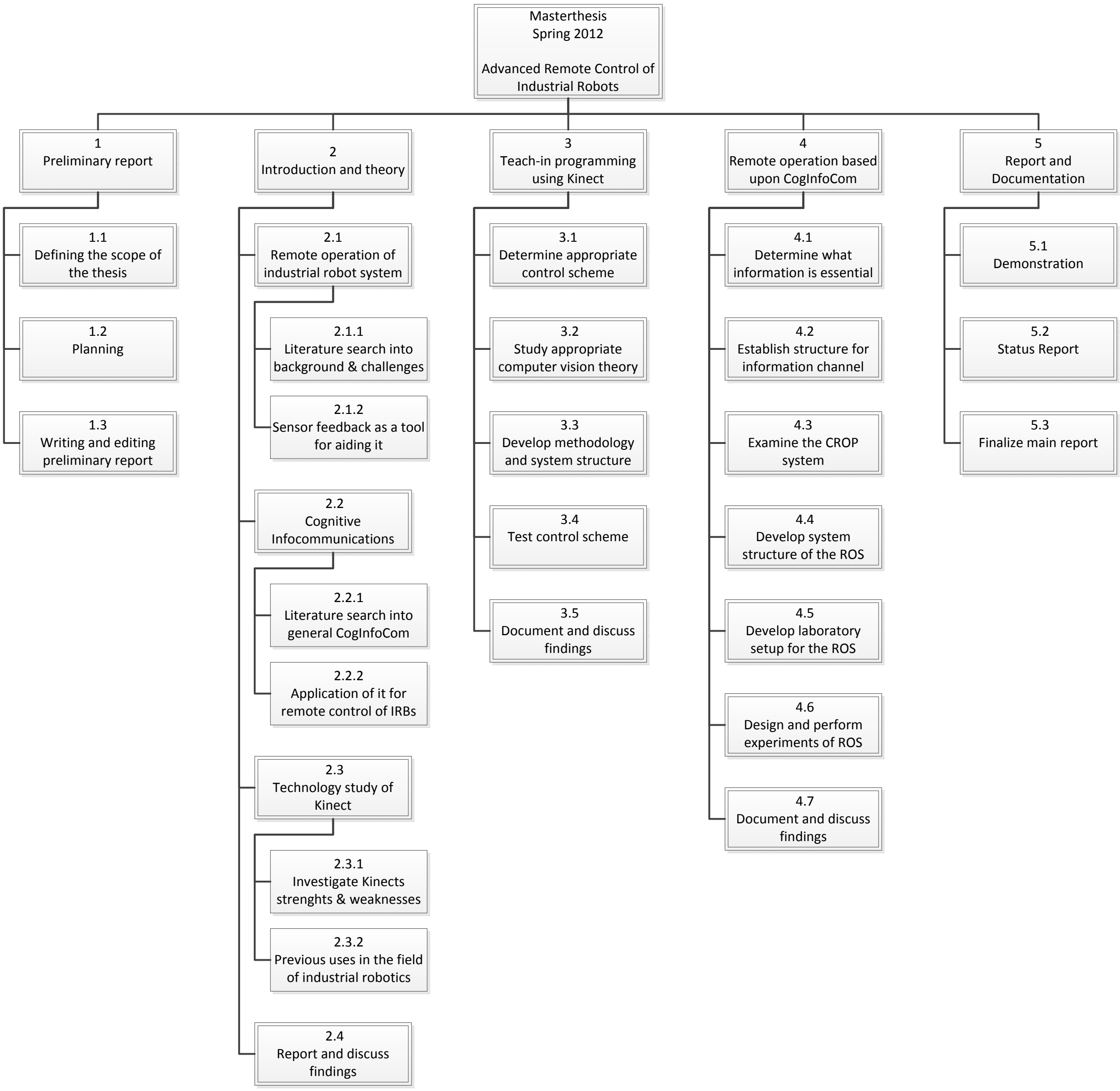
Conditions:

- The candidate can get sufficient knowledge about the subject.
- The candidate can cooperate with his supervisor and other people involved during the work.
- The project is sufficiently planned.

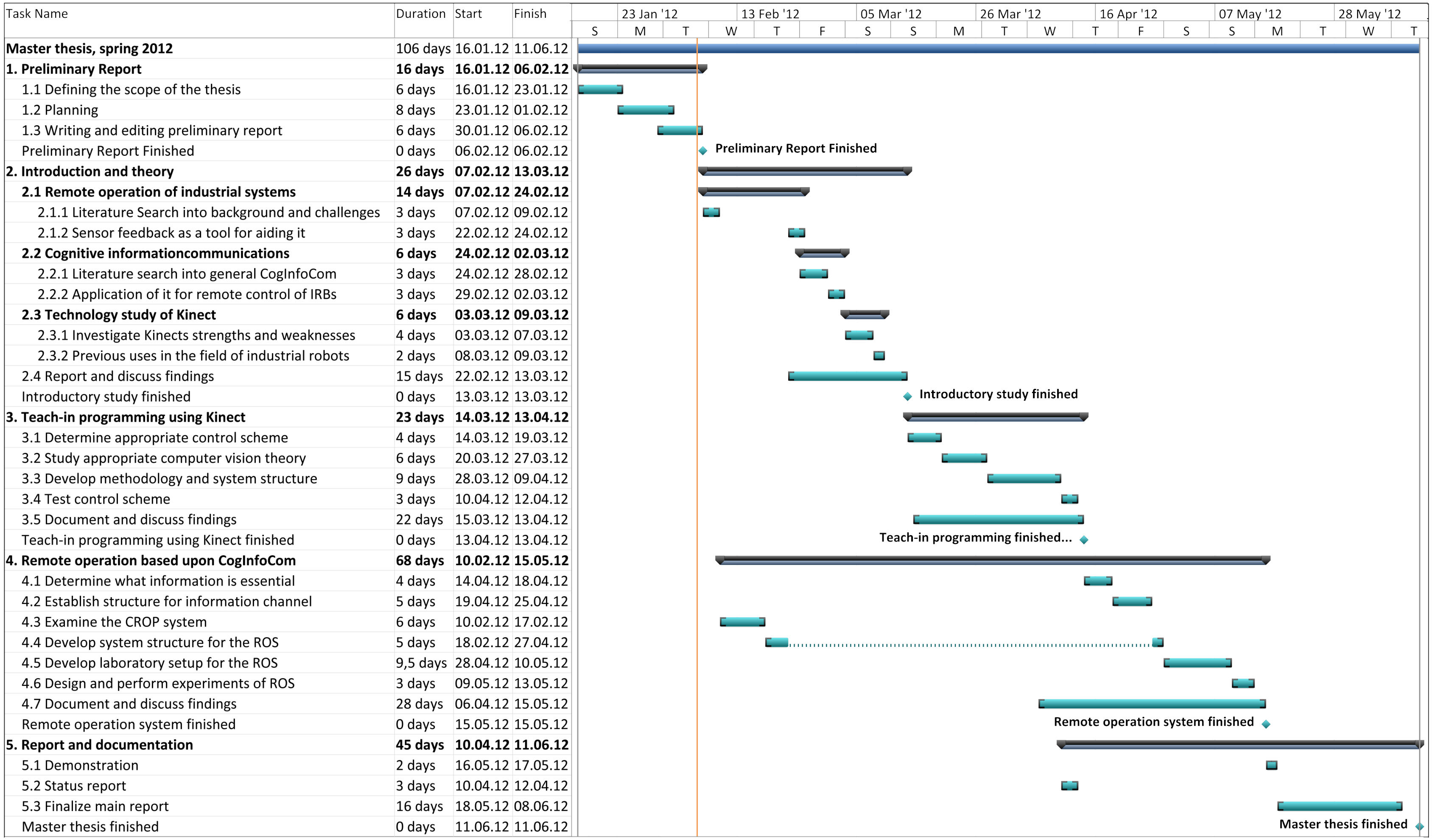
Risks:

- Illness
- The magnitude of the tasks are too comprehensive.

3.2 Work breakdown structure



3.3 Gantt



Task Name	Duration	Start	Finish
Master thesis, spring 2012	**106 days**	**16.01.12**	**11.06.12**
1. Preliminary Report	**16 days**	**16.01.12**	**06.02.12**
1.1 Defining the scope of the thesis	6 days	16.01.12	23.01.12
1.2 Planning	8 days	23.01.12	01.02.12
1.3 Writing and editing preliminary report	6 days	30.01.12	06.02.12
Preliminary Report Finished	0 days	06.02.12	06.02.12
2. Introduction and theory	**26 days**	**07.02.12**	**13.03.12**
2.1 Remote operation of industrial systems	**14 days**	**07.02.12**	**24.02.12**
2.1.1 Literature Search into background and challenges	3 days	07.02.12	09.02.12
2.1.2 Sensor feedback as a tool for aiding it	3 days	22.02.12	24.02.12
2.2 Cognitive informationcommunications	**6 days**	**24.02.12**	**02.03.12**
2.2.1 Literature search into general CogInfoCom	3 days	24.02.12	28.02.12
2.2.2 Application of it for remote control of IRBs	3 days	29.02.12	02.03.12
2.3 Technology study of Kinect	**6 days**	**03.03.12**	**09.03.12**
2.3.1 Investigate Kinects strengths and weaknesses	4 days	03.03.12	07.03.12
2.3.2 Previous uses in the field of industrial robots	2 days	08.03.12	09.03.12
2.4 Report and discuss findings	15 days	22.02.12	13.03.12
Introductory study finished	0 days	13.03.12	13.03.12
3. Teach-in programming using Kinect	**23 days**	**14.03.12**	**13.04.12**
3.1 Determine appropriate control scheme	4 days	14.03.12	19.03.12
3.2 Study appropriate computer vision theory	6 days	20.03.12	27.03.12
3.3 Develop methodology and system structure	9 days	28.03.12	09.04.12
3.4 Test control scheme	3 days	10.04.12	12.04.12
3.5 Document and discuss findings	22 days	15.03.12	13.04.12
Teach-in programming using Kinect finished	0 days	13.04.12	13.04.12
4. Remote operation based upon CogInfoCom	**68 days**	**10.02.12**	**15.05.12**
4.1 Determine what information is essential	4 days	14.04.12	18.04.12
4.2 Establish structure for information channel	5 days	19.04.12	25.04.12
4.3 Examine the CROP system	6 days	10.02.12	17.02.12
4.4 Develop system structure for the ROS	5 days	18.02.12	27.04.12
4.5 Develop laboratory setup for the ROS	9,5 days	28.04.12	10.05.12
4.6 Design and perform experiments of ROS	3 days	09.05.12	13.05.12
4.7 Document and discuss findings	28 days	06.04.12	15.05.12
Remote operation system finished	0 days	15.05.12	15.05.12
5. Report and documentation	**45 days**	**10.04.12**	**11.06.12**
5.1 Demonstration	2 days	16.05.12	17.05.12
5.2 Status report	3 days	10.04.12	12.04.12
5.3 Finalize main report	16 days	18.05.12	08.06.12
Master thesis finished	0 days	11.06.12	11.06.12

3.4 Work packages

Work Package	
WBS Number: 1	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Preliminary report	**Work Place:** IPK
Title: Preliminary Report	
Description of Work: This report will give a summary of all the problems that are to be solved. It will describe the goals and limitations and an analysis of the problems in the thesis, as well as a detailed plan for the work. The work plan shall contain a Work Breakdown Structure (WBS) which will lead into a Gantt chart. The Gantt chart will also include an estimate of the workload of each work packages. The Milestones for the work should be pointed out in the Gantt chart. The Preliminary report should also include a description of the work methodology. The due date for the preliminary report is February 6th 2012 and is to be handed in to the supervisor.	
Planned Start: 16.01.12	**Planned Finish:** 06.02.12
Revision number: V0.1	**Date:** 01.02.12

Work Package	
WBS Number: 2.1	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Introduction and theory	**Work Place:** IPK
Title: Remote operation of industrial systems	
Description of Work: This work package is meant to give the candidate insights into the background of why there is a need for advanced remote control of industrial robots. It will contain literature search into how small and medium sized enterprises (SMEs) differ from bigger enterprises and its implications in terms of their need for knowledge transfer and the challenges associated with serving this need. In addition, general tele robotics theory will be studied in order to give the candidate a better overview of the technical side of remote operation systems. Studies into how sensors feedback can be of use for simplifying and/or assisting a remote operation system and its operator will also be investigated. Special focus will be aimed at force/torque- and vision sensors.	
Planned Start: 07.02.12	**Planned Finish:** 24.02.12
Revision number: V0.1	**Date:** 01.02.12

Work Package

WBS Number: 2.2	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Introduction and theory	**Work Place:** IPK
Title: Cognitive informationcommunications (CogInfoCom)	
Description of Work: In this set of work packages, the candidate is to investigate the contents of the field of CogInfoCom. It will entail a literature search with focus on how it can be utilized in the field of advanced remote control of industrial robots, proper semantics of the field and structure of CogInfoCom information channels. Special focus will be set on how the teachings of CogInfoCom can be applied to the available hardware and software at PPM AS.	
Planned Start: 24.02.12	**Planned Finish:** 02.03.12
Revision number: V0.1	**Date:** 01.02.12

Work Package

WBS Number: 2.3	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Introduction and theory	**Work Place:** PPM
Title: Technology study of Kinect	
Description of Work: The purpose of this work package is to give the candidate understanding and insights into the strengths and weaknesses of the Kinect, as well as how it has previously been used in the field of industrial robotics. The candidate will also explore the performance of the Kinect with special focus on its depth-precision. In connection to this, the candidate may design and perform practical experiments.	
Planned Start: 03.03.12	**Planned Finish:** 09.03.12
Revision number: V0.1	**Date:** 01.02.12

Work Package

WBS Number: 3.1	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Teach-in Programming using Kinect	**Work Place:** PPM
Title: Determine appropriate control scheme	
Description of Work: The candidate is to determine an appropriate control scheme on the basis on what has been learned by the work packages 2.1-2.3. The candidate is to consider control schemes implications in terms of CogInfoCom (WBS 2.2) as well as the technical feasibility learned from the technology study of the Kinect (WBS 2.3). Primary control scheme to be investigated: Perform direct teach-in programming by translating and rotating either the operators hand or an object (e.g. an analog to the physical robots end-effector). Considerations into methods for including additional functionality such as a method for signaling actions to the physical system (e.g. gripping) should also be included.	
Planned Start: 14.03.12	**Planned Finish:** 19.03.12
Revision number: V0.1	**Date:** 01.02.12

Work Package

WBS Number: 3.2	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Teach-in Programming using Kinect	**Work Place:** PPM
Title: Study appropriate computer vision theory	
Description of Work: The motive of this work package is to give the candidate insights into the needed computer vision theory for developing the chosen control scheme (WBS 3.1). The candidate is to search for both theory and potential software packages required to tackle the challenges associated with the control scheme. The candidate may also search for potential supporting technologies such as gyros, accelerometers or other sensors if it is deemed necessary and/or advantageous for realizing the control scheme.	
Planned Start: 20.03.12	**Planned Finish:** 27.03.12
Revision number: V0.1	**Date:** 01.02.12

<table>
<tr><th colspan="2">Work Package</th></tr>
<tr><td>WBS Number:
3.3</td><td>Project:
Advanced Remote Control of Industrial Robots</td></tr>
<tr><td>Subproject:
Teach-in Programming using Kinect</td><td>Work Place:
PPM</td></tr>
<tr><td colspan="2">Title:
Develop methodology and system structure</td></tr>
<tr><td colspan="2">Description of Work:
In the work of developing the methodology and system structure, the candidate is to consider how to reduce complexity, ensuring easier implementation into the larger remote operation system.

It should include among others:
A way of determining that the operator is ready to control the robot via the Kinect, ensuring that there are no unwanted actions sent to the robot system.
Methods for detecting false data (e.g. large jumps during tracking etc.)
Possibility to tune various parameters of the control scheme, e.g. 1:1 control versus a tuned down version of 1:0.5.
Possibility to include additional data (such as gripper command etc.)</td></tr>
<tr><td>Planned Start:
28.03.12</td><td>Planned Finish:
09.04.12</td></tr>
<tr><td>Revision number:
V0.1</td><td>Date:
01.02.12</td></tr>
</table>

<table>
<tr><th colspan="2">Work Package</th></tr>
<tr><td>WBS Number:
3.4</td><td>Project:
Advanced Remote Control of Industrial Robots</td></tr>
<tr><td>Subproject:
Teach-in Programming using Kinect</td><td>Work Place:
PPM</td></tr>
<tr><td colspan="2">Title:
Test control scheme</td></tr>
<tr><td colspan="2">Description of Work:
The purpose of the work is to survey the control scheme in terms of user friendliness and performance.

Examining user friendliness may consist of testing the control scheme on untrained personnel, instructed to program the robot, and collect feedback.

In performance testing, the focus shall be on perceived precision as well as its robustness. Performance testing in terms of response time/speed will not be investigated unless there is very pronounced lag between the controlling operator and the robot.</td></tr>
<tr><td>Planned Start:
10.04.12</td><td>Planned Finish:
12.04.12</td></tr>
<tr><td>Revision number:
V0.1</td><td>Date:
01.02.12</td></tr>
</table>

Work Package

WBS Number: 4.1	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Remote operation based on CogInfoCom	**Work Place:** PPM
Title: Determine what information is essential	
Description of Work: The work is intended to map out what information that has to be communicated between the remote and physical system. This could include; joint angles, state of emergency switch, tool definition etc. In addition to mapping out what data is appropriate to be communicated, the candidate is to investigate how this data can be obtained (e.g. how to establish the state of the emergency switch?). Appropriate placement of cameras and other sensing devices should also be considered.	
Planned Start: 14.04.12	**Planned Finish:** 18.04.12
Revision number: V0.1	**Date:** 01.02.12

Work Package

WBS Number: 4.2	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Remote operation based on CogInfoCom	**Work Place:** PPM
Title: Establish structure for information channel	
Description of Work: The work should result in appropriate handling of information going between the remote and physical system in terms of how the data is to be structured in virtual channels. It should be able to give some information precedence over others (e.g. if an emergency signal comes, other data may be less important).	
Planned Start: 19.04.12	**Planned Finish:** 25.04.12
Revision number: V0.1	**Date:** 01.02.12

Work Package

WBS Number: 4.3	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Remote operation based on CogInfoCom	**Work Place:** PPM

Title:
Examine the CROP system

Description of Work:
Background:
As previously mentioned, PPM AS has a system called CROP (Cognitive remote operation). Much of the research will revolve around restructuring and expanding this system.

László Nagy, is one of people behind developing and implementing the CROP system at PPM AS. He will visit PPM AS from 17.02 until 22.02 in order to set up the existing system as well as give insights into how the system can be expanded upon and restructured. This workpackage has therefore been placed before his arrival in order for the candidate to get sufficient overview of the system for effective cooperation with László.

The candidate is to investigate the CROP systems structure and investigate how it can be reused and expanded upon for using the Olimex high-speed interface and other wanted functionality.

Planned Start: 10.02.12	**Planned Finish:** 17.02.12
Revision number: V0.1	**Date:** 01.02.12

Work Package

WBS Number: 4.4	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Remote operation based on CogInfoCom	**Work Place:** PPM
Title: Develop system structure for ROS	
Description of Work: The work entailed in this work package is dependent on the outcome from work package 4.3. The work is to be done in two periods; the first while László Nagy is at PPM AS to set up and introduce the CROP systems structure (17.02 – 22.02), and one when all new functionalities are defined. The work will consist of determining how the system can be structured use the Olimex high speed interface, the new control scheme and informationhandling. For restructuring for using the Olimex, the following has to be taken into account: • How to handle the Nachi MR20s kinematics • Trajectory generation • Position saving • Programming robot movement. Also, investigations has to be done into how the inclusion of the developed teach-in programming method and changes regarding information handling etc. should be done.	
Planned Start: 18.02.12	**Planned Finish:** 27.04.12
Revision number: V0.1	**Date:** 01.02.12

Work Package

WBS Number: 4.5	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Remote operation based on CogInfoCom	**Work Place:** PPM
Title: Develop laboratory setup for the ROS	
Description of Work: This work contains the practical development and implementation of the chosen methodology and structure for the system. The system is to include all essential functionality to be able to perform experimental testing of the remote operation system. The scope and magnitude of this work package is highly dependent of the findings during work package 4.3 and 4.4.	
Planned Start: 28.04.12	**Planned Finish:** 10.05.12
Revision number: V0.1	**Date:** 01.02.12

Work Package

WBS Number: 4.6	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Remote operation based on CogInfoCom	**Work Place:** PPM
Title: Design and perform experiments for ROS	
Description of Work: The focus of this work is to develop experiments that can reflect real industrial task and operating conditions. The experiments are to be performed either from an external location or by introducing realistic delay between the operator and the physical system.	
Planned Start: 09.05.12	**Planned Finish:** 13.05.12
Revision number: V0.1	**Date:** 01.02.12

Work Package

WBS Number: 5.1	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Report and documentation	**Work Place:** PPM
Title: Demonstration	
Description of Work: In order to give a better and clearer view of the resulting system, a demonstrationvideo(s) will be made. This should give the viewer a clearer understanding of both how the experiments and laboratory is set up, as well as the real life usage and performance of the system and control scheme.	
Planned Start: 16.05.12	**Planned Finish:** 17.05.12
Revision number: V0.1	**Date:** 01.02.12

Work Package

WBS Number: 5.2	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Report and documentation	**Work Place:** PPM

Title:
Status report

Description of Work:
The work will reveal the progress so far, and give an updated work plan. It will discuss any delays and changes to the original plan.

If there are changes to the original work plan, these are to be approved by the supervisor.

The status report is to be handed in to the supervisor during the thesis work as well as included as an appendix in the main report.

Planned Start: 10.04.12	**Planned Finish:** 12.04.12
Revision number: V0.1	**Date:** 01.02.12

Work Package

WBS Number: 5.3	**Project:** Advanced Remote Control of Industrial Robots
Subproject: Report and documentation	**Work Place:** IPK & PPM

Title:
Finalize main report

Description of Work:
The main report should be presented as a scientific report, thus edited as such.

The candidate is to write on the main report during work on the various parts of this thesis (work packages; 2.4, 3.5 and 4.7), this work package is therefore intended to include all finalization of the main report such as closing discussions, results and conclusions, as well as final proof-reading.

The main report is due June 11th, and two bound copies as well as an electronic (.pdf) copy is to be delivered.

Planned Start: 18.05.12	**Planned Finish:** 08.06.12
Revision number: V0.1	**Date:** 01.02.12

3.5 Milestones

Event	**Date**
Preliminary report finished	06.02.12
Introduction and theory finished	13.03.12
Teach-in programming using Kinect finished	13.04.12
Status Report	14.04.12
Remote operation system based on CogInfoCom finished	14.05.12
Master thesis delivered	11.06.12

Preliminary Literature Index

Baranyi, P. and Csapo, A. (2010). Cognitive infocommunications: Coginfocom. In *Computational Intelligence and Informatics (CINTI), 2010 11th International Symposium on*, pages 141–146. IEEE.

Csapo, A. and Baranyi, P. (2011). A unified terminology for CogInfoCom applications. In *Cognitive Infocommunications (CogInfoCom), 2011 2nd International Conference on Cognitive Infocommunications*, pages 1–6. IEEE.

Galambos, P. and Baranyi, P. (2011). Vibrotactile force feedback for telemanipulation: Concept and applications. In *Cognitive Infocommunications (CogInfoCom), 2011 2nd International Conference on Cognitive Infocommunications*, pages 1–6. IEEE.

HUNOROB. http://www.itm-lab.org/academic-research-and-r-d-programs/hunorob-program.

Nagy, I. (2011). Cognitive aspects of augmented reality applications. In *Cognitive Infocommunications (CogInfoCom), 2011 2nd International Conference on Cognitive Infocommunications*, pages 1–3. IEEE.

Rolstad, A. r. (2006). *Praktisk prosjektstyring*.

Siciliano, B. and Khatib, O., editors (2008). *Springer Handbook of Robotics*.

Sobota, B., Hrozek, F., Korecko, S., and Szabo, C. (2011). Virtual reality technologies as an interface of cognitive communication and information systems. In *Cognitive Infocommunications (CogInfoCom), 2011 2nd International Conference on Cognitive Infocommunications*, pages 1–5. IEEE.

Thomessen, T. and Kosicki, T. (2011). Cognitive audio-visual infocommunication applied in remote support for industrial robot systems. In *Cognitive Infocommunications (CogInfoCom), 2011 2nd International Conference on Cognitive Infocommunications*, pages 1–5. IEEE.

Thomessen, T., Nagy, L., and Olah, K. (2011). Remote control of industrial robot systems using cognitive info communication. In *Cognitive Infocommunications (CogInfoCom), 2011 2nd International Conference on Cognitive Infocommunications*, pages 1–6. IEEE.

(Csapo and Baranyi, 2011) (Nagy, 2011) (Thomessen and Kosicki, 2011) (Thomessen et al., 2011) (Galambos and Baranyi, 2011) (Sobota et al., 2011) (Siciliano and Khatib, 2008)

Appendix J

Progress Report

NTNU – Trondheim
Norwegian University of
Science and Technology

Progress Report

Advanced Remote Control of Industrial Robots

Stud.techn. Fredrik Reme

Spring - 2012

Department of Production and Quality Engineering
Norwegian University of Science and Technology
in Partial Fulfillment of the requirements
for the Degree of Master of Science

Supervisor: Professor II Trygve Thomessen

1 Present Status

At the time of writing (1. May, 2012), the candidate has hit the following milestones in reference to the original plan from the pre-study report:

- Teach-in Programming Finished
- Remote Operation System Finished

In more detail, these entails:

Teach-in Programming: The candidate has determined an appropriate control scheme for use during remote operation, developed its methodology and system structure. At the highest system-levels, the end result can be explained as follows. The human operator moves a gripper analog in space. The gripper analogs 6DOF is determined using a Kinect for position and a magnetic-, angular rate-, gravity-sensor (MARG) for its attitude. This data is consequently interpreted and applied to the robot in various ways depending on which control mode the human operator chooses.

Remote Operation System: While the teach-in programming scheme is the core for the direct control of the robots movements, an overarching system that enables the operator to efficiently control the robot remotely has been developed. The main mentality behind the system, is to present various important information in such a way that it feels intuitive for the operator. The resulting system includes, among others, camera displays, robot and environments simulation as well as employing CogInfoCom techniques for effectively communicating the active control modes, nearing singularities, gripper status etc. Initial testing of the remote operation system has also been performed with satisfying results.

Current and remaining work:
In addition to reaching the aforementioned milestones, the candidate is at this time in the process of documenting the implemented system, determining appropriate experiments and methodology for testing the systems performance, as well as stipulate the final frame for the final report.

2 Work Plan

The candidate has kept the original work plan presented in the pre-study report with regard to the higher level work packages. In hindsight, the work plan could have been more high-level, as planning of the lower-level work packages seems to have resulted in them being a bit skewed in terms of relevancy and/or time allocated. Therefore these has been slightly changed in content and/or in allocated time during the work.

Schedule Discrepancies
With regard to the high-level and general work packages, the main discrepancies between the original plan and the actual progress during the thesis mainly concern work packages for documenting and reporting findings during the work. This was deemed ineffective for the candidate and slowed down the work. The result of this is that the candidate has finished many of the work packages regarding implementation etc. ahead of schedule, while some work packages for documentation and reporting remains.

Due to relatively few deviations in the plan and actual progress, the candidate has deemed it unnecessary to make an updated gantt diagram for the remainder of the work.