Quality-Latency Trade-Off in Bilateral Teleoperation

Victor Millnert

LUND UNIVERSITY

Department of Automatic Control

MSc Thesis
ISRN LUTFD2/TFRT--5951--SE
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Media-Tryck
Lund 2014

Abstract

The purpose of this thesis is to investigate how the latency in mobile networks affect the quality of highly demanding and sensitive applications running on it. Furthermore, this thesis will provide some information to what is going on in the field of Cloud Computing and the Internet of Things. It will hopefully spark a discussion about what possibilities will come with the development of the Cloud and Internet of Things.

The application chosen was a bilateral teleoperation, with force feedback, controlled in 6 dimensions. To investigate how the quality depends on network latency, different network models were simulated as the communication channel. The networks chosen to be simulated were a 3G, 4G, and a 5G cellular network along with a wired network chosen as a baseline.

On this setup two main experiments were done. The first one was a collision test and the second one a dexterity test, where a user was supposed to pick up a small wooden brick and put it into a box. The results from the experiments showed that there was indeed a difference in behavior when having a network delay larger than 20 ms.

Acknowledgments

I would like to express my gratitude to all the wonderful people who have helped me with this thesis. First I would like to thank my two advisers Anders Robertsson and Johan Eker. Anders, from the Dept. Automatic Control, has been very patient in teaching me everything needed to complete this thesis. His understanding for nearly everything is truly inspiring and while managing a workload suitable for three persons he somehow always finds the time to help whenever needed. Johan, who is both with the Dept. Automatic Control and with Ericsson, has also been truly inspiring in providing visions and ideas along with a great sense of direction. Whenever I spoke with my advisers I always left them with a smile on my face, along with a feeling that I would probably need three years to do everything we wanted to do. I would also want to thank Mahdi Ghazaei for helping me with lots of smaller questions and understanding. Finally I would like to thank everyone at the department for providing such inspiring and stimulating environment, not neglecting Eva Westin for giving me much needed access to the coffee room.

Thank you all!

Contents

List of Figures

1

Introduction

In a very near future the digital world and the analog world will converge into one. Texas Instruments predicts that every person on the planet will be able to program and control their environment by themselves by using smart sensors, vast computing resources with fast, reliable Internet every where and all the time [Texas Instruments, 2013]. It will spark an amazing collaboration between smart machines, embedded systems and people allowing us to finally realize the full potential of the Internet, [Harbor Research, 2014]. The convergence in mind is the one between Cloud Computing, the Internet of Things and the Fifth Generation mobile networks that together will combine infinite computing resources and smart real-time analysis with data from sensors collecting data from all aspects of our lives and in the industry brought together by the ability to have a reliable connection everywhere and all the time.

1.1 The Future is Cloudy

There is no escaping that we are seeing a tremendous growth in Cloud computing these days. The concept has finally reached the critical mass and almost everyone have heard about the Cloud, but far from everyone knows how massive the explosion really is. The Cloud Computing industry is growing faster than ever. According to Cisco two-thirds of the workloads will be processed in the Cloud by 2017 [Cisco, 2013a]. They also predict that we will see a 5-fold increase in the Cloud IP-traffic over the next five years. A research group at UC Berkeley say Cloud computing will be the new utility in the future, comparing it with electricity, gas and the telephone, [Armbrust et al., 2009].

Werner Vogel at Amazon Web Services says that the Cloud is a major game changer in how businesses are working [Malik, 2013]. By eliminating the need for IT infrastructure we will see small businesses scale faster and in new areas of the world, making it possible for the ones in developing countries to leapfrog and start competing with the western world.

Some examples of the amazing things happening today include IBM's cognitive technology, Watson. Being a part time chef cooking and creating never-before seen recipes and winning Jeapordy — once thought as impossible for a computer [IBM, 2014b]. If that's not enough Watson is helping one of the world's most respected medical facilities, MD Anderson, to conduct real-time analysis of unstructured data in patient records, [IBM, 2014a].

The elasticity and scalability with the pay-as-you-go concept of Cloud computing is what made Pfizer use Amazon Web Services (AWS). It enables them to explore deep scientific questions with short bursts of peak activity without having to pay for the infrastructure needed [AmazonWebServices, 2014].

Driving Factors of Cloud Computing are among others the evolution of connected devices. Some of the driving forces on the growth of Cloud computing are the increasing amount of devices connected to the Internet and the rapidly increasing amount of data available for processing.

1.2 Sensors will watch your every step

When more and more smart things and sensors are starting to connect to the Internet we will see the Internet of Things. Today we have about 5 billion connected things, Texas Instruments predict that by 2020 there will be 50 billion and within our lifetime a trillion, [Texas Instruments, 2013]. IDC predicts that, if the major hurdles in the development of the Internet of Things can be overcome, there will be over 200 billion connected devices by 2020, resulting in a trillion dollar industry, [Dignan, 2013].

While today we might be able to control different things with with our smartphone, there is one app for one thing. And we have to pick up and interact with our smartphone to control our environment. The next step, imagined by T.I, is to not having to do so. Imagine checking into a hotel. Before you arrive the hotel room is on standby, cool, and dark. Once you arrive you automatically get a secure key app on your phone, the hotel room starts heating up to your preferred levels. When you approach the door it automatically unlocks and the room starts playing the music you like, or reads your emails to you or puts on the news. It knows that you soon have dinner plans and want to take a shower so it prepares a bath for you and calls a cab to pick you up just in time to arrive at the restaurant [Texas Instruments, 2013].

One of the major hurdles for the development of Internet of Things is connectivity. To be able to connect different things running on different OS using different standards and protocols. IFTTT.com is a small step on the way, allowing a user to manually program his or her different things through a "If This Then That" interface, [IFTT, 2014]. However, the need is greater, and there will have to be some really simple way to connect all the things together for the every-day user. Another major hurdle will be the latency of the things connected. Cisco says that reducing

latency to and from the Cloud will be absolutely necessary for developing advanced services, [Cisco, 2013a].

One of the major players in the connectivity business, Ericsson, is working on this, [Pretz, 2013]. They are, together with IEEE, developing standards with the goal of application-independent tools allowing for many-to-many logical connectivity. Along with this Ericsson has the goal of unlimited access to information and sharing of data being available anywhere and anytime to anyone and anything - which they call the 5G Radio Access, [Ericsson, 2013].

1.3 The feedback-loop and purpose of this thesis

It is natural to see that the Cloud, the IoT and the Network industries are growing in some form of symbiosis, stimulating each other and converging into One. This is where the amazing will happen. Harbor Research puts it in a very nice way: "*It will bend the traditional linear value chain into a "feedback loop"*", [Harbor Research, 2014].

The first purpose of this thesis is to stimulate a discussion about where we are going with the development of the Cloud, IoT and Networks. A discussion perhaps leading to new ways of controlling things or what will be needed to reach the convergence point. Will it be possible to move the control algorithms to the Cloud, or will it have to be closer to the processes?

The focus, however, will be a very small part of this. This thesis will focus on how latency affects the quality of an advanced application run over a mobile network. The application of choice is a bilateral teleoperation. If it would be possible to run a high-quality and reliable bilateral teleoperation over the mobile network the end-result could be amazing. One of the main applications would be in remote surgery. It would allow a surgeon to receive force-feedback while performing a remote surgery. In turn, this would provide better care at remote areas of the world as well as at temporary emergency cites. It would allow for one surgeon to operate on three different patients, at three different small cities, on three different continents on the same day!

2

Background

This chapter will cover some background material needed for this thesis. The first three sections, Cloud Computing, Internet of Things, and Mobile Communications, is meant to paint a picture of the rapid growth in these fields. These developments act as the intellectual basis for the problem formulation, which is described at the end of this chapter. Before the problem formulation, however, are two technical sections covering bilateral teleoperation, and robot kinematics, which were needed to develop the application used in this thesis.

2.1 Cloud Computing

This year, 2014, Amazon Web Services is expected to make a revenue of $5 billion. That is a 50% increase from last year's $3.1 billion, making it the fastest growing software business in the history, [Vance, 2014]. In 2015 they are expected to make a profit of $6.7 billion. The Cloud computing is growing in symbiosis with the evolution of the Internet of Things and together with the ability to connect everything everywhere there are amazing things to come. In Copenhagen they are using an array of sensors in the street-light fixtures as part of a network. It enables them to help with traffic control, identify carbon dioxide levels and detect when the garbage cans are full [Cardwell, 2014].

What is the Cloud?

The Cloud concept has been around for quite some time now, and previously it has been quite hard to define exactly what it means. However, the various definitions are starting to converge more and more. Gartner, [Gartner IT Glossary, 2014], defines the Cloud as:

> Cloud computing is a style of computing where scalable and elastic IT-enabled capabilities are delivered as a service to external customers using Internet technologies.

[Buyya et al., 2009] provides the following definition:

> A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s) based on service-level agreements established through negotiations between the service provider and the customer.

Amazon Web Services says that the main features of Cloud computing will be the flexibility, scalability, elasticity and reliability it provides to the end-user, [AmazonWebServices, 2013].

[Armbrust et al., 2009] and [Buyya et al., 2009] provides a great overview of what Cloud computing is. They say it is important to note that a Cloud is neither a cluster nor a grid. That a cluster is a collection of inter-connected computing machines working together as a single computing resource. A cluster consists of common computers in the scale of hundreds where one usually needs to be a member in order use the shared resource. A grid is a parallel and distributed system consisting of high-end computers and usually in the scale of thousands of servers. Furthermore, they say that the main difference of the Cloud is that it runs several VMs (Virtual Machines) on the physical machines, making it possible to allocate resources in a very dynamic manner. This next-generation method of "virtualized" nodes (virtualized storage and computing) has been made possible through the hypervisor technologies. The VMs make it possible to run several independent application using different operating systems on the same physical machine.

The Cloud can be divided into four parts: The User/Broker, The SLA Resource Allocator, the VMs and the physical machines.

- **Brokers/Users** submits the requests to use a particular service provided by the Cloud.
- **SLA Resource Allocator** acts as the interface between the Data Centers and the Brokers/Users.
- **VMs** is the virtual machines running on the physical machines. Every VM is completely isolated from the other VMs and may concurrently run different applications on the same physical machine.
- **Physical Machines** is the physical machines that makes the data center.

How to use the Cloud

A number of companies are working towards connecting the cloud with the Internet of Things, i.e. with sensors and actuators. Emerging applications include vast sensor arrays connected to data analysis in the Cloud. Two major players working in this direction are General Electric with there GE Industrial Internet:

> "GE’s development of the Industrial Internet is connecting Intelligent Machines, Advanced Analytics and People at Work High frequency

> real-time data brings a whole new level of insight on system operations. Machine-based analytics offers yet another dimension to the analytic process. The combination of physics-based approaches, deep sector specific domain expertise, more automation of information flows, and predictive capabilities can join with the existing suite of “big data” tools. The result is the Industrial Internet, which encompasses traditional approaches with newer hybrid approaches that can leverage the power of both historic and real-time data with industry specific advanced analytics." [Evans and Annunziata, 2012]

and Cisco with their Cisco Fog:

> "The distinguishing Fog characteristics are its proximity to end-users, its dense geographical distribution, and its support for mobility. Services are hosted at the network edge or even end devices such as set-top-boxes or access points. By doing so, Fog reduces service latency, and improves QoS, resulting in superior user-experience"[Cisco, 2013b]

Studying the field of Cloud computing one quickly realizes that it is highly competitive, many different players are trying to distinguish themselves from the rest. However, one also realizes that several of them are pushing towards the same thing; converging into one solution that allows real-time analysis of massive amounts of data to help optimize processes and distribution as well as to aid in difficult decision making.

2.2 Internet of Things

While IoT might be one of the driving forces of Cloud Computing, the Cloud architecture is driving the IoT evolution. Along with the development of new, low-cost and energy-efficient ways of enabling wireless communication we see a rapid growth of the number of connected things.

With the increased connectivity and the lowered cost of RFID chips the Internet will have more and more devices and "things" connected to it. This will expand the communication over the Internet to include human-human, human-thing, and thing-thing interactions, the Internet of Things. For a more thorough survey of the promise and future of the Internet of Things the reader is referred to [Munjin and Morin, 2012], [Coetzee and Eksteen, 2011] and [Tan and Wang, 2010].

What is the Internet of Things?

Though the general picture of the IoT is that there will be lots and lots of sensors and "things" connected over the Internet Texas Instruments put it in a very nice way:

> "The IoT creates an intelligent, invisible network fabric that can be sensed, controlled and programmed. IoT-enabled products employ embedded technology that allows them to communicate, directly or indirectly, with each other or the Internet." [Texas Instruments, 2013]

Some of the main driving forces behind the Internet of Things are some specific networks:

- Machine-to-Machine networks
- Vehicle-to-Vehicle networks
- Personal medical networks
- Home entertainment systems
- Home automation/security systems
- Personal fitness networks
- Highway sensor networks
- Energy management networks

These different networks will transform the way we interact with our environment. It will transform everything from optimizing our factories, prevent car accidents, and detect strokes and heart attacks before they occur.

Whats the catch?

What about the hurdles that will have to be overcome before we can get there? One of the biggest is that currently the devices are very vertical, one app for one thing, and they are all using different protocols and unable to "glue" together in an easy manner. There will have to be a substantial amount of work in developing easy-to-use standards making it possible to connect different clusters of things and intuitively program them.

Another issue is that all the different sensors and things will have to be more or less self-sustaining in regards of power. They will need to be extremely energy-efficient and it will most likely have to be possible to charge them in a wireless manner.

However, the absolutely biggest hurdle is security. Having devices recording data of every aspect of every person and every factory will undoubtedly be a major target for ill-intended actions. [William Boldt, 2014] says that this authentication issue will be the *sine qua non* ("without which there is nothing") of the IoT.

2.3 Mobile Communications

The huge growth of the Cloud industries and the development of the IoT is putting a lot of pressure on the telcom industry. The big question is how the mobile networks will have to evolve in order to survive. Ericsson is doing a serious amount of research in this area. Their vision:

> "Unlimited access to information and sharing of data available anywhere and anytime to anyone and anything"[Ericsson, 2013]

They call it the 5G Radio Access and they are hoping to reach standardization within a few years. If so, they can start rolling out the 5G capabilities around 2020. This will be the backbone of the convergence between the Cloud and IoT.

The long-term outcome of this, they predict, will be the networked society. It will not mean a new technology replacing the precious, 4G and 3G, but rather one that extends them. It will be a Cloud that is more fluid, dynamic and responsive to the service needs of the end-users. The goal is to cut the Cloud free from the anchors, or physical data centers, and open it up for innovation.

2.4 Bilateral Teleoperation

The first bilateral teleoperation was built by Goertz in the mid ’40s, [Hokayem and Spong, 2006]. Since then a lot of research has been made in order to improve the stability and transparency of the teleoperation link. Teleoperation allow an operator to use a joystick, *master device*, in order make a robot, *slave device*, interact with a distant environment. If the slave device have the possibility of sensing forces, these can be reflected back to the master device giving the operator force-feedback. Such link is called a *bilateral* teleoperation link, see Fig. 2.1. Transparency is a measure of how easy it is for the operator to interact with the environment. One can say that the higher the transparency the greater teleprecence the operator feel. Achieving a high transparency generally conflicts with having a high stability of the system.

Bilateral teleoperation is very useful since it allows an operator to scale the motions or forces, while interacting with the distant environment, making it easier to perform complex tasks or to interact with heavy objects. Another promising area is within the telesurgery field, since it would allow the surgeon to feel the contact forces of the surgical robot. Furthermore, it would provide for great simulation and training for the aspiring surgeons.

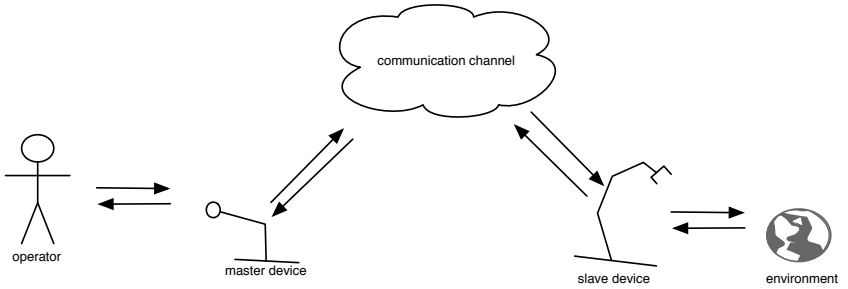

Figure 2.1 Overview of a bilateral teleoperation link.

Modeling of a Bilateral Teleoperation

A more formal way of modeling can be seen in Fig. 2.2. Here the operator transmits the desired velocity to the master-device while receiving force feedback. The slave device tries to move according to the desired velocity while measuring forces and sending them back to the master-device.

Figure 2.2 A very useful way of depicting bilateral teleoperation.

The stability, quality and transparency of a bilateral teleoperation is highly dependent on the communication channel. The reason is that communication delay may cause the system to be unstable.

2.5 Robot Kinematics

> "Listen, Josef," the author began, "I think I have an idea for a play." "What kind," the painter mumbled (he really did mumble, because at the moment he was holding a brush in his mouth). The author told him as briefly as he could. "Then write it," the painter remarked, without taking the brush from his mouth or halting work on the canvas. The indifference was quite insulting. "But," the author said, "I don't know what to call these artificial workers. I could call them Labori, but that strikes me as a bit bookish." "Then call them Robots." Source: [Capek and Wyllie, 2010]

This is how the word *robot* came about. How it really happened is hard to say but it is famously known that Karel Capek introduced the term to the public through his play "Rossum's Universal Robots" in 1920. Since then the research of robotics has evolved into a broad field covering a range of topics including industrial robots, mobile robots, humanoid robots, medical robots and bionics to name a few [Garcia et al., 2007].

In order to accurately control a robot, one needs a good mathematical representation of it. Since this thesis only covers serial-link robots a short introduction to this will be covered. For a more in-depth introduction to robotics modeling and control the reader is referred to two very well written books, [Corke, 2011] and [Spong et al., 2005].

Rigid Motions and Transformations

A robot arm is represented by a series of links that are connected through joints. The configuration, or the pose, of the robot is determined by the states of these joint-angles. Different joint-angles result in different positions of its end-effector. The robot treated in this thesis have 6 degrees-of-freedom, meaning its end-effector can change rotation (3 DOF) as well as translation (3 DOF).

Before describing how the end-effector and the joint-angles are related a way to represent rigid motions and orientations needs to be established. Throughout this thesis the 4x4 *homogeneous transformation matrix*, T, will be used. It provides a very intuitive way of describing points and poses in different coordinate systems. We write the transformation matrix in the three-dimensional case as

$$T = \begin{pmatrix} R_{3x3} & t_{3x1} \\ 0_{1x3} & 1 \end{pmatrix} \tag{2.1}$$

where R is a three-by-three rotational matrix and t is a one-by-three translation matrix. The rotation matrix R can be defined as

$$R = R_x(\theta_x) R_y(\theta_y) R_z(\theta_z) \tag{2.2}$$

where R_x, R_y and R_z are rotations about the x-, y- and z-axis, respectively. The translation matrix t is defined as

$$t = \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix} \tag{2.3}$$

If a 4x4 transformation matrix, T_B^A, relates frame $\{B\}$ to frame $\{A\}$, one can transform points originally expressed in frame $\{B\}$, p^B, into points expressed in frame $\{A\}$, p^A, through

$$p^A = T_B^A p^B \tag{2.4}$$

where

$$T_B^A = \begin{pmatrix} R_B^A & t \\ 0_{1x3} & 1 \end{pmatrix} \tag{2.5}$$

Note that p^A and p^B is the same point, just expressed in two different coordinate frames. Using the same reasoning it is easy to relate poses, consisting of both a rotation and a translation, between different coordinate systems.

Since it is quite easy to get lost while doing transformations, a good way to illustrate their relationship is needed. Through out this thesis a map like the one shown in Fig. 2.3 will be used to illustrate this.

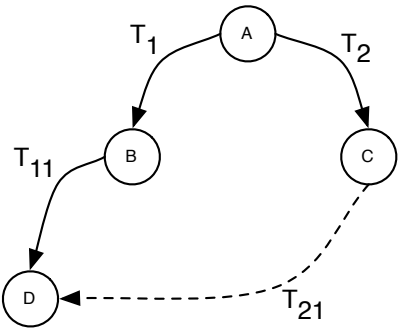

Figure 2.3 A useful way to illustrate the relationships between different transformations.

Referring to Fig. 2.3, we say that

$$T_B^A = T_1 \tag{2.6}$$

and that

$$T_D^A = T_B^A \cdot T_D^B = T_1 \cdot T_{11} \tag{2.7}$$

Using the inverse of the transformation matrix one can easily start to express rather complex relations, for instance, T_D^C can be expressed as

$$T_D^C = \left(T_C^A\right)^{-1} \cdot T_B^A \cdot T_D^B = {T_2}^{-1} \cdot T_1 \cdot T_{11} = T_{21} \tag{2.8}$$

Using this, it's very straight forward to express poses in the different coordinate frames. For instance a pose, ξ^C, expressed in frame $\{C\}$ can be transformed and instead expressed in frame $\{D\}$ as ξ^D through

$$\xi^D = T_{11}^{-1} \cdot T_1^{-1} \cdot T_2 \cdot \xi^C \tag{2.9}$$

Forward Kinematics

A systematic approach to express the geometry of a serial-link robot arm was presented in [Hartenberg and Denavit, 1964], and is commonly known as the Denavit-Hartenberg convention, (DH). It provides a clever way of assigning coordinate frames to the different links so that the 4x4 homogeneous transformation matrix relating coordinate frame $\{i-1\}$ to $\{i\}$ only depends on the parameters θ_i, d_i, a_i and α_i. For a revolute joint only θ_i is a variable while the others are fixed. The meaning of these parameters are explained in Table 2.1.

θ_i	angular rotation around z_{i-1}-axis
d_i	distance along the z_{i-1}-axis between the two x-axes
a_i	distance between the two z-axes
α_i	angular rotation about x_i-axis relating the two z-axes

Table 2.1 Short description of the Denavit-Hartenberg parameters.

Using this, the transformation matrix, relating frame $\{i-1\}$ and frame $\{i\}$, can be expressed as

$$A_i^{i-1} = R_z(\theta_i)\, T_z(d_i)\, T_x(a_i)\, R_x(\alpha_i) \tag{2.10}$$

and expanded as

$$A_i^{i-1} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\cos\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\sin\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & & 1 \end{pmatrix} \tag{2.11}$$

For a robot with n links the transformation matrix relating the end-effector pose, ξ, and the base frame can be expressed as

$$\xi \backsim T_E^0 = A_1^0 \cdot A_2^1 \cdots A_n^{n-1} \cdot T_E^n \tag{2.12}$$

and in our case where all joints are revolute ξ is variable depending on the different joint angles and can be expressed as

$$\xi = \kappa(q) \tag{2.13}$$

where q is the vector of the joint-angles.

Inverse Kinematics

Inverse kinematics handles the case when one has the pose of the end-effector and wishes to derive the joint angles. Different robots will need different methods of deriving this. However in the case where the robot has a spherical wrist, i.e., the axes of the three last wrist joints intersect in a single point, there exists a closed-form

solution for deriving these joint-angles. Most industrial robots have such wrists, and indeed the ABB IRB-140 which was used in this thesis. In any other case, and for a more in-depth description, the reader is once more referred to [Corke, 2011] and [Spong et al., 2005].

When the end-effector pose ξ is known the joint angles can be expressed as

$$q = \kappa^{-1}(\xi) \tag{2.14}$$

Velocity Kinematics

While it is possible to control a robot's motion by making it move through an incremental range of positions it is not desirable. Instead one wishes to control the motion through a combination of desired position along with a desired velocity. The most common way of expressing the relationship between the end-effector velocity and the joint velocity is through the *Jacobian*, J. The relationship can be expressed as

$$v = J(q)\dot{q} \tag{2.15}$$

where $v = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$ is the angular and translational velocity of the end-effector and q and $\dot{q}$ are the joint angles and joint velocities, respectively. The Jacobian is a $6 \times n$-matrix where n is the number of links in the robot arm. For a robot arm with only revolute joints and when using the Denavit-Hartenberg convention the Jacobian can be expressed as

$$J = \begin{bmatrix} z_0 \times (p_1 - p_0) & \cdots & z_{n-1} \times (p_n - p_{n-1}) \\ z_0 & \cdots & z_{n-1} \end{bmatrix} \tag{2.16}$$

Note that z_i is the vector that joint i revolutes about and that $(p_{i+1} - p_i)$ is the vector form joint i to joint $i+1$.

In the case where the Jacobian is a square matrix the joint velocities can be derived through the inverse of Eq. (2.16).

$$\dot{q} = J^{-1}(q)v \tag{2.17}$$

To derive the instantaneous velocity of the end-effector, consider two poses, $T_e^C(t)$ and $T_e^C(t+\delta t)$, both expressed in frame $\{C\}$, see to Fig. 2.4. Then one can express the differential transformation matrix as

$$\delta T_e^C(t) = T_e^{-1}(t) \cdot T_e(t+\delta t) \tag{2.18}$$

and expanded, the differential transformation matrix can be expressed as

$$\delta T_e^C(t) = \begin{pmatrix} 1 & -\omega_z & \omega_y & v_x \\ \omega_z & 1 & -\omega_x & v_y \\ -\omega_y & \omega_x & 1 & v_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \delta t \tag{2.19}$$

which can be used to derive the velocity vector v used in Eq. 2.16 and Eq. 2.17.

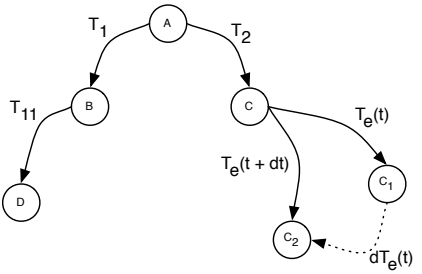

Figure 2.4 Differential transformation matrix.

2.6 Problem Formulation

The focus of this thesis is to investigate and characterize how big the network latency could be if a highly demanding application were to be run over a mobile network in the future. The outcome will hopefully provide requirements and use cases for the ongoing 5G development.

In order to find what a suitable network latency should be, a highly demanding application is designed in this thesis. The application chosen is a bilateral teleoperation, as described above. Different networks will be simulated, with different latency distributions, in the communication channel of the chosen application to investigate how they affect the quality. With this setup, three different experiments will be made. One will focus on a collision, one will focus on the dexterity of the application, and one will focus on the stability of the application.

3

Method

This chapter will describe the setup and solutions used to build the application in this thesis, the bilateral teleoperation. It begins by describing the solution used to make the application stable despite delays in the communications channel. Following this, there is a short section about how the different networks were modeled and simulated. Finally, there are two sections describing the hardware setup and the software setup, along with the different algorithms used.

3.1 Wave Variable

Inspired by the phenomena of the passive waves, which are stable despite the occurrence of time-delay, the *wave variable* was presented in [Niemeyer and Slotine, 1991]. The idea was that instead of sending velocity in one direction and force in the other, a mixture of these were to be sent in both directions. By choosing these wave variables in a clever manner it is possible to guarantee a passive communication channel and thus a stable system. These waves can be modeled as

$$u_m = \dot{x}_m - F_m \qquad v_m = \dot{x}_m + F_m \tag{3.1}$$

$$u_s = \dot{x}_s + F_s \qquad v_s = \dot{x}_s - F_s \tag{3.2}$$

Since a lossless wave might become a standing wave, some impedance b is added to the system. The new system can be seen in Fig. 3.1 and expressed as

$$u_m = b\dot{x}_m - F_m \qquad v_m = b\dot{x}_m + F_m \tag{3.3}$$

$$u_s = b\dot{x}_s + F_s \qquad v_s = b\dot{x}_s - F_s \tag{3.4}$$

The storage function and its stored energy, P, can the be expressed as

$$P = \frac{1}{4b}\left(u_s^2 - v_s^2 + u_m^2 - v_m^2\right) \tag{3.5}$$

Furthermore, using the fact that the output of one side will simply be input of the other side, with a small time-delay T.

$$v_s(t) = u_m(t-T) \tag{3.6}$$
$$v_m(t) = u_s(t-T) \tag{3.7}$$

Thus the storage function can thus be expressed as

$$\begin{aligned} P &= \frac{1}{4b}\left(u_s^2(t) - u_s^2(t-T) + u_m^2(t) - u_m^2(t-T)\right) \\ &= \frac{1}{4b}\frac{d}{dt}\int_{t-T}^{t}\left(u_s^2(\tau) + u_m^2(\tau)\right)d\tau \end{aligned} \tag{3.9}$$

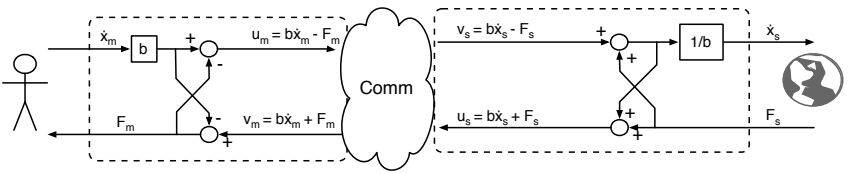

Figure 3.1 Bilateral teleoperation using wave variables.

3.2 Simulating Latency

The approach to simulate the latency of different networks was chosen in order to be able to evaluate the effects in a very controlled manner. However, to get accurate and realistic simulations there was a need for accurate and realistic models of the different networks. The four different networks chosen to be simulated were

- Wired Network
- 5G Network
- 4G Network
- 3G Network

To be able to create the different network models, there was first some measurements of the 3G, and 4G network latency. This data were then fitted with suitable

distributions. The model for the 5G network was based on the model for the 4G network, but with a lower mean latency and a lower variance.

The communication channel of the bilateral teleoperation was a wired network. Obviously, there was a small latency for this network. The simulation of the different network models was simply made by adding a delay to the already existing wired network.

3.3 Hardware Setup

The Robotics Lab at the Department of Automatic Control, Lund Institute of Technology, provides a great possibility of controlling industrial robots through a series of program built at the department. An overview of the different modules and how they are connected can be seen in Fig. 3.2. This chapter will go through the different modules in turn and explain their task.

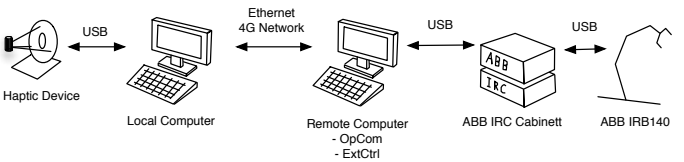

Figure 3.2 Overview of the hardware setup and how the different modules are connected.

The Haptic Device

The haptic device used in this setup is an Omega 7, built by *Force Dimension*. It provides 7 degrees of freedom, three for translation, three for rotation and one for gripping. For a more detailed version of the specifications, see [Force Dimension, 2014]. The Omega 7 haptic device also provides force feedback in three dimensions. The force-feedback is limited to 12 N for the translation case and ± 8 N for the grasping tool. A picture of the Omega 7 can be seen in Fig. 3.3.

Figure 3.3 Omega 7 haptic device, copyright of ForceDimension

Local Computer

The local computer used in this setup is a Linux distribution, running on Fedora 16. In order to interact with the haptic device a series of commands are provided by *Force Dimension* through a library. The *C* programming language is used to build the interface and the connection to the haptic device. A more detailed description of that software will be covered in Sec. 3.4. The haptic device is connected to the local computer via a USB cable.

Communications

The connection between the local and remote computer is made through the local area network. In order to make it possible to send data between different, often incompatible, versions of software a binary protocol written by the Department of Automatic Control is used [Blomdell et al., 2010]. The protocol is called LabComm and is very easy to use. One does just have to define a description file with the data signals that are to be sent and then compile it. The compiler generates the programs necessary the establish a connection between the programs using different software. Since LabComm is a one-way communications protocol it is possible to store the data stream to a file for later analysis. For a more detailed description of LabComm, please see [Blomdell et al., 2010].

Remote Computer and IRC Cabinet

On the remote computer a Simulink-generated controller is running, which is connected to a two-way protocol called ORCA, which is built on top of the LabComm protocol, [Blomdell et al., 2005]. This makes it possible to send data streams to the Simulink controller from the local computer, as well as to connect to the IRC cabinet, which is the robot computer provided by ABB. Thus making it possible to read the signals and trajectories generated by the IRC cabinet's main computer and send new updated ones, generated by the Simulink-generated controller, to the IRC axis

computer. See Fig. 3.4.

Through an operator interface (OpCom) it is possible to enter the submit mode and the obtain mode, shown in in Fig. 3.4, as well as to change predefined parameters in the Simulink controller. When the submit mode is active, data is read from the IRC main computer and delivered to the Simulink controller through the *irb2ext* sources. These signals can then be logged, manipulated, and sent out to the robot through the *ext2irb* sinks. However, the OpCom have to be set in *obtain* mode for this to work. Furthermore it is possible to predefine new LabComm signals in the Simulink controller, making it possible to send data from external devices, such as a force sensor or a haptic device. The remote computer is running Xenomai for the real-time requirements [Xenomai, 2014].

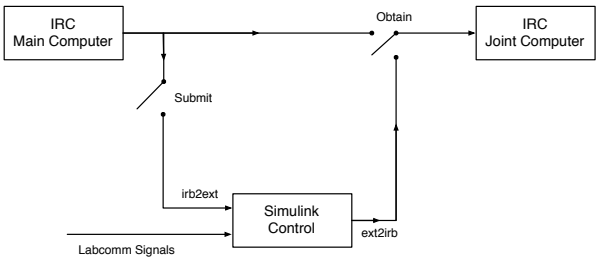

Figure 3.4 Schematic for the extended control structure between the Simulink controller and the IRC cabinet.

The Robot

The robot used in this setup is an ABB 140 industrial robot with 6 degrees-of-freedom. It is equipped with a 6 DOF force sensor and a gripper tool.

3.4 Software Setup

The bilateral teleoperations setup consists of two local controllers, one using *C* code acting as the interface towards the haptic device and one using Simulink/Matlab acting as the interface towards the robot. In order to simulate delay in the communication channel some buffers on the master side were used. These where able to simulate delays with standard distributions close to what was measured in Sec. 5.1. An overview of what the two interfaces cover of the teleoperation can be seen in Fig. 3.5.

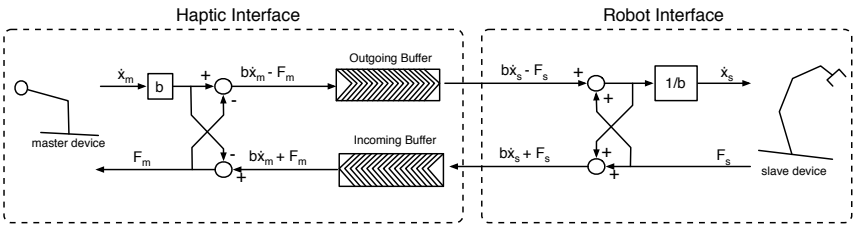

Figure 3.5 An overview of the two different interfaces. The Haptic interface handles the communication with the Omega.7 as well as simulating network delays while the Robot interface only interacts with the robot.

Kinematic Chain

In order to achieve an intuitive control of the robot through the haptic device a few different coordinate frames have to be defined. For the haptic device these include a *Base Frame*, *Task Frame*, *Handle Frame*, and a *Tool Frame* . In the same manner the robot utilizes a *Base Frame*, *Task Frame*, *Flange Frame*, *Sensor Frame*, and a *Tool Frame*.

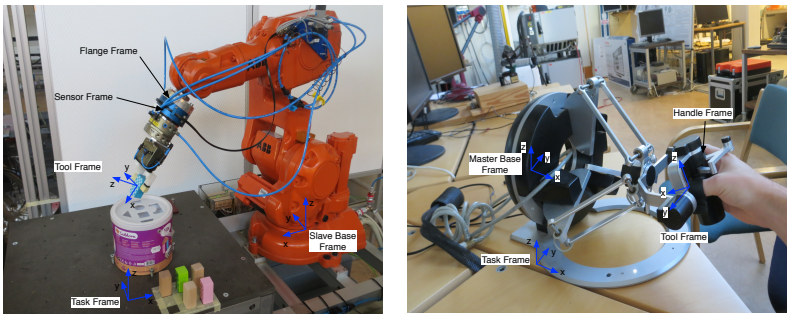

Figure 3.6 Left image shows the defined coordinate frames for the robot. The right image shows the ones for the haptic device.

The Task frame is defined as the space in which the operator wishes to manipulate the environment and objects. The Task frame does not have to be fixed as it is in our application. But rather it can be dynamic and for instance follow a moving object, such as a different robot arm.

The Sensor frame is defined when setting up the force sensor and building the robot application. It represent the orientation of the force sensor so that the measured and estimated forces can accurately be transferred to a different coordinate

frame. The Sensor frame is fixed to the Flange frame, which is defined internally in the robot.

The Tool frame for both the robot and the haptic device are defined by the software developer when building the application. Figure 3.6 illustrates where these coordinate frames are defined for the haptic device (right image) and the robot (left image), respectively.

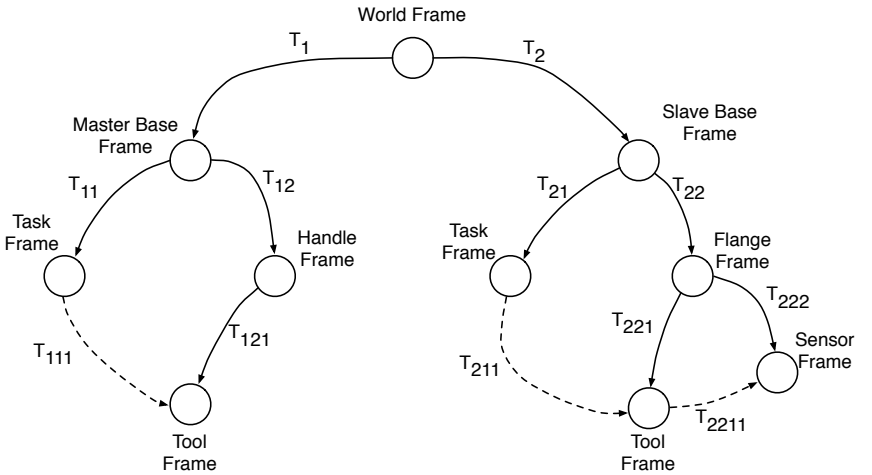

Figure 3.7 Kinematic chain of the bilateral teleoperation.

The whole kinematic chain for the haptic device and the robot is illustrated in Fig. 3.7. It shows the relationship between the different coordinate frames and defines the transformation matrices that relates them. In order to achieve a control of the robot manipulator that feels natural for the operator the relationship between the Task frame and the Tool frame for both the robot and the haptic device must satisfy Eq. (3.10). It states that the relationship between the Task- and Tool frame of the haptic device must be the same as that between the Task- and Tool frame of the robot. Since T_{111} and T_{211} cannot be measured directly they have to be calculated using Eqs. (3.11) and (3.12), utilizing measurements of T_{21}, T_{121}, T_{22}, and T_{221}.

$$T_{111} = T_{211} \tag{3.10}$$

$$T_{111} = T_{11}^{-1} \cdot T_{12} \cdot T_{121} \tag{3.11}$$

$$T_{211} = T_{21}^{-1} \cdot T_{22} \cdot T_{221} \tag{3.12}$$

Since the robot manipulator, or Tool frame, is chosen to be controlled through velocity references the velocity of the master Tool frame has to be calculated and is done so according to Sec. 2.5 and Eqs. (2.18) and (2.19). From there the angular and translational velocities, $v_x, v_y, v_z, w_x, w_y, w_z$, are sent to the robot. On the robot side the new Tool frame velocity is then used to calculate the new velocity of the flange and then of the motor joints which in turn is integrated in order to receive the reference positions for the motor joints. A more thorough walk-through of this will be provided in Sec. 3.4. In order to calculate the new Tool velocity as well as to transform the forces sensed by the force sensor to the Tool frames the following calculation was used

$$T_{2211} = T_{221}^{-1} \cdot T_{222} \tag{3.13}$$

Haptic Interface

The haptic interface has three purposes. The first one is to talk to the haptic device, send force commands, and read orientations and positions. The second one is to be the local controller and to calculate the different wave variables being sent to the robot and received from the robot. Its last purpose is to simulate network delays so that the quality of the application can be evaluated during different delays. An overview of the haptic interface can be seen in Fig. 3.8.

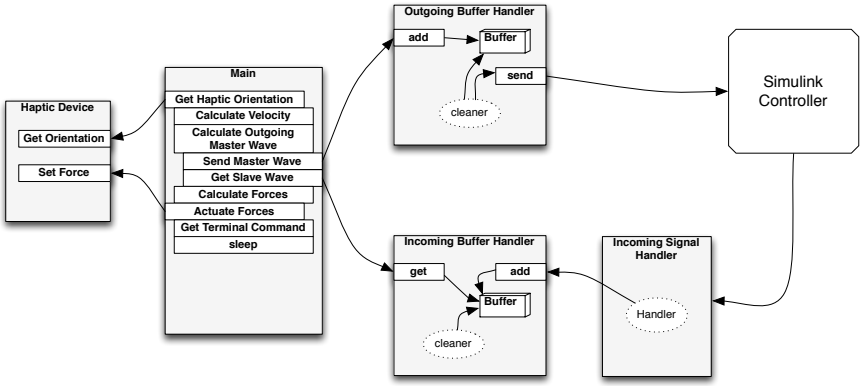

Figure 3.8 Overview of the haptic interface. It uses one main thread, one for handling the outgoing buffer and two for handling the incoming signals as well as the keeping the incoming buffer neat.

Main Thread The pseudo code for the algorithm running in the main thread of the haptic interface is described in Algorithm 3.4. The period was chosen to be 4 ms since the Simulink controller was implemented using that period. The operator can,

through different keyboard commands, choose which network delay to simulate. It is possible to simulate a wired network (no added latency), a 5G network, a 4G network, and a 3G network. Furthermore the operator has the option to hit 'SPACE' which gives the opportunity to move the haptic device without moving the robot, i.e. not recording any signals from the haptic device. This is great when running out of workspace at the haptic device or when the operator needs to reorient the haptic device.

In order to give the operator the possibility of gripping objects with the robot manipulator the distance between the haptic devices finger and the thumb was measured. If the gripper was closed a close-signal was sent to the robot interface and if the gripper was open an open-signal was sent.

Algorithm 1 Pseudo code for the main thread of the haptic interface.

```
while We Should Run do
    Read the haptic position and orientation
    Calculate the differential velocity
    Calculate the outgoing wave variable
    if We Should Simulate Network Delay then
        Send the outgoing wave variable to the outgoing buffer
        Get the incoming wave variable from the incoming buffer
    else
        Send the outgoing wave variable to the Simulink controller
        Get the newest incoming wave variable
    end if
    Calculate the force that is to be actuated on the haptic device
    Send the force to the haptic device
    Read any commands from the operator
    Sleep - for the remainder of the period
end while
```

Buffers When calling the function to add a signal to the buffer a payback time is calculated using the built-in C++ functions *std::normal_distribution* and *std::extreme_value_distribution*. The cleaner, see Fig. 3.8, then makes sure to keep the correct signal first in the buffer (for the incoming buffer) and to send the signal at the correct playback-time (for the outgoing buffer). The added delays are based on the measurement results described in Sec. 5.1. The 5G network is simply chosen to bridge the gap between the wired network and the 4G network, using a distribution similar to that of the 4G network. The latency simulated for each buffer (one-way delay) can be seen in Table 3.1.

Network	Distribution	μ [ms]	σ [ms]
3G	General Extreme Value	149.4	12.5
4G	Normal	30	2.5
5G	Normal	10	1

Table 3.1 Distributions used to simulate the network delays. 3G uses a general extreme value distribution while 4G and 5G uses a normal distribution.

Robotic Interface

The robotic interface has two purposes, acting as the local controller and calculating the wave variable to send to the haptic interface, see Fig. 3.5. The incoming signals to the robot interface are described in Table 3.2 and the outgoing signals in Table 3.3. The robot is controlled through the two signals *ext2irb.posRef* and *ext2irb.velRef* which are sent to the low level computer in the IRC cabinet, see Sec. 3.3. The Simulink controller in the robotic inteface can be divided into three parts, *calculate wave variables*, *calculate force for the Tool* and *calculate new motor joint reference values*.

Incoming Signals	Description
irb2ext.robot[0].joint[i].posRef	Initial motor joint position [motor rad]
irb2ext.robot[0].joint[i].velRef	Initial motor joint velocity [motor rad/s]
irb2ext.robot[0].joint[i].posRawAbs	Measured motor joint position [motor rad]
jr3_comedi[i]	Force measurements from the force sensor [N]
masterWave[i]	Wave variable sent from the haptic interface

Table 3.2 Incoming signals to the robot interface along with their description.

Outgoing Signals	Description
ext2irb.robot[0].joint[i].posRef	Motor joint reference position [motor rad]
ext2irb.robot[0].joint[i].velRef	Motor join velocity reference [motor rad/s]
slaveWave[i]	Wave variable sent to the haptic interface
ext2irb.robot[0].joint[i].mocgendata.value1	Signal to open the gripper
ext2irb.robot[0].joint[i].mocgendata.value2	Signal to close the gripper

Table 3.3 Outgoing signals from the robot interface along with their description.

Calculate Wave Variables The simulink model for calculating the wave variable can be seen in Fig. 3.9. As input it takes the received wave variable from the master device, or haptic interface along with the force readings transformed into the Tool frame. Some scaling is necessary in order to make sure that both inputs have equal impact on the two outgoing signals, the wave variable to be sent to the haptic interface and the Tool reference velocity, respectively.

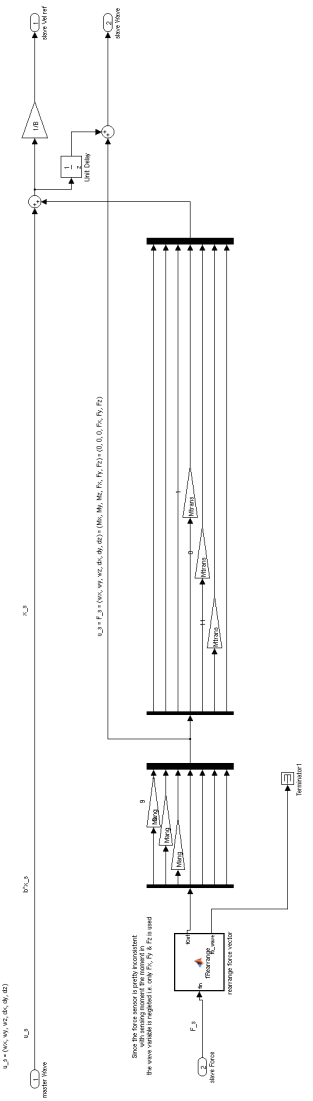

Figure 3.9 Simulink block for calculating the wave variable. Incoming signals are the force in the Tool frame and the wave variable sent from the haptic interface. Outgoing signals are the wave variable to be sent to the haptic interface and the reference velocity of the robot Tool.

Calculate Tool Force Since the force sensor is not located at the tip of the robot manipulator, or tool, some force transformations have to be made. First the force readings goes through some force compensations, to eliminate any constant or dynamic offset due to the tool attached to the robot manipulator. After that the force have to be converted from the sensor location to the tool location. To do this there is a Simulink block for force transmission in the *Extctrl* library developed by the department. After the force has been transformed to the tool frame it is passed through a low-pass filter as well as a small dead-zone, which is an interval around zero where the forces are set to zero. The reason being to minimize high-frequency noise and small oscillations around zero. The Simulink model can be seen in Fig. 3.10.

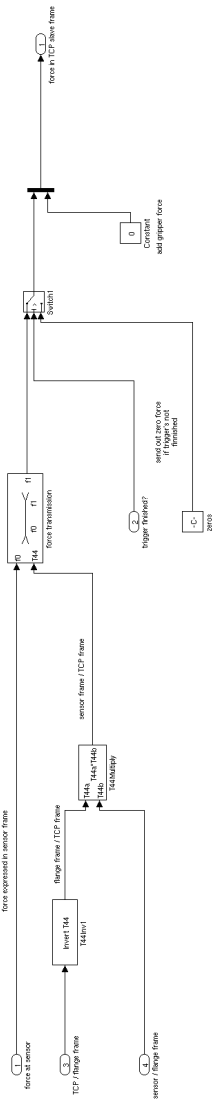

Figure 3.10 Simulink model for the force transmission calculations.

Calculate Motor Joint Reference Values The new Tool reference velocity, which was calculated in the wave variable block, is passed through a low-pass filter to eliminate high-frequency noise. Then, after scaling the translational velocity from mm/s to m/s, the reference is passed through a safety block to ensure the safety of both the robot and its environment. The safety block checks the following:

- If the robot manipulator is outside the work space
- If the robot manipulator velocity is too high
- If the forces acting on the robot is too high

After the safety check the Tool reference velocity is used to calculate the desired motor joint velocities. The Simulink model can be seen in Fig. 3.11. Input to the Simulink block is *Tool reference velocity*, *transformation matrix from flange frame to Tool frame*, *the measured motor joint values*, and *the initial motor joint values*.

First it calculates the desired velocity of the flange. Next, using the Jacobian, it calculates the desired velocity of the robot arms and the desired velocity of the robot joints. The new reference velocity is then passed through a low-pass filter and saturated so they do not exceed any dangerous limits. The two outputs of the Simulink block are both motor joint velocities. One is used as the reference velocity, which is sent to the IRC cabinet, and the other one is integrated, with Forward Euler method, to retrieve the desired motor joint positions to send to the IRC cabinet.

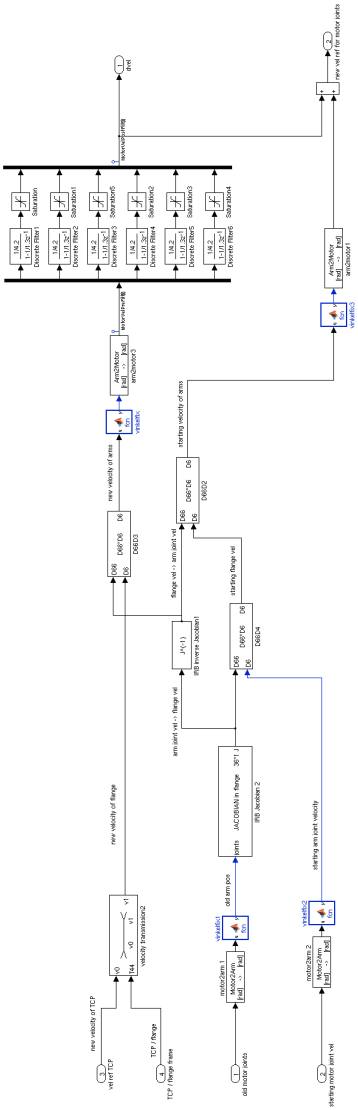

Figure 3.11 Simulink model for calculating the reference velocity of the motor joints.

4

Experiments

This chapter will describe the different experiments made in this thesis. The first one was to measure the network latency for the 3G and 4G networks. The data collected from this experiment was used to create models for the simulation of the four different network models. The three other experiments were made on the bilateral teleoperation in order to investigate how the different network models affected the stability and quality of the system.

4.1 Latency Measurements

The approach used to measure the latency occurring over the 3G network and the 4G network was to connect two different computers using two different network dongles. The dongles used were made by Huawei and enabled 3G and 4G network traffic. They also allowed the user to force traffic over a desired network, say the 4G network. In order to measure the latency over the two networks data packets was sent between the computers once every second for 1800 seconds. The round-trip delay was then recorded. For an illustration of the setup see Fig. 4.1.

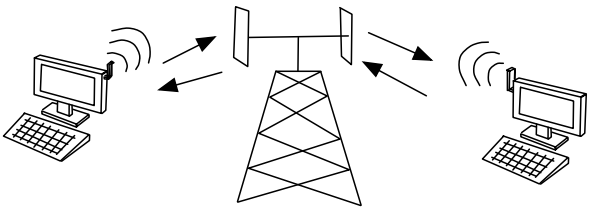

Figure 4.1 Setup used to measure the latency over the 3G and 4G networks.

4.2 Collision Test

The collision test was made by forcing a collision between the robot manipulator and a hard surface. The surface chosen was a brittle foam board. The surface of the foam board was not elastic, so there was no spring effect in the collision behavior. However, if the collision force would be too great the foam-board surface would collapse and prevent an accident. The collision test was made four times, once for each of the simulated networks. During the collision, forces, positions, and velocities were recorded.

4.3 Dexterity Test

The dexterity test used in this experiment was inspired by the Purdue Pegboard Test, where the test subject is supposed to put different pegs in different holes, and the Minnesota Manual Dexterity Test, where the test subject is supposed to move small pucks into different holes. Both of those tests are timed in order to find if there is a significant difference in the amount of items the test subject can put in place at a specific time, or if the is a difference in the time it takes to complete putting a specific number of items in place. The test subject in our experiment was set to pick up small bricks made out of wood and put them in a rectangular hole, see Fig. 4.2. There was a total of six bricks and through controlling the robot, the test subject would try to put them in the box as quick as possible. The total time of the test was recorded in order to spot a difference between the different network modes. There was a total of five test runs on each network mode, all made by the same test subject.

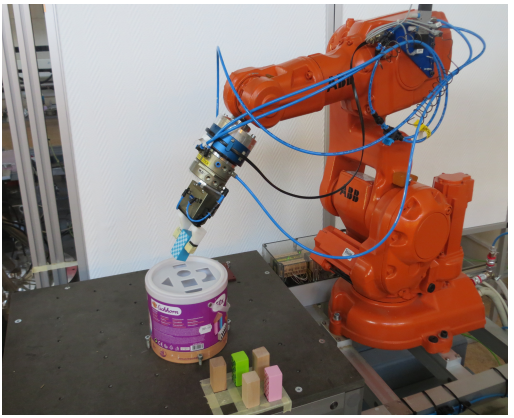

Figure 4.2 Test setup for the dexterity test.

Stability Test

In order to further investigate if there are any differences in the quality between the different network modes there was yet another test. This test was made to test the stability of the system and done so by forcing the robot to collide the brick with the edges of the hole. The test was made by having the operator picking up a brick, finding the hole and putting the brick in the hole, then, without releasing the brick, the operator would force a collision with the edge of the hole. During this test, forces, positions, and velocities were recorded.

5

Results

This chapter covers the results for the different experiments. The first section shows the measurement results of the 3G and 4G network latency. It also shows the different distributions that were fitted with the collected measurement data. Following the measurement results is a section showing the simulated latency for the four different networks. The last three sections cover the results of the collision test, the dexterity test, and the stability test.

5.1 Latency Measurements

The results for the latency measurements of the 3G network can be seen in Fig. 5.1. The plots show the round-trip latency recorded for each of the packets. The measurement lasted for about 1800 seconds sending one packet every second. In the upper image there are two distinct bands, one around 150 ms and one around 300 ms. Furthermore there are a lot of high-latency outliers accompanying the low-latency bands. The reason for having two different bands were unknown and not further investigated since the purpose was to find some data about the magnitude of the 3G latency. Therefore before trying to fit a suitable distribution to the 3G network latency the outliers were excluded only to include the most dense middle-band, with a latency around 300 ms, leaving out any data point above 600 ms or below 250 ms. The result of the new latency is seen in the lower image of Fig. 5.1.

The 3G network latency was fitted with a suitable distribution using the Matlab tool *'dfittool'*. The result was a general extreme value distribution (GEV) and can be seen in Fig. 5.2. The image shows the cumulative distribution function (CDF) along with the collected latency data for the 3G network. The GEV distribution can be expressed according to Eq. 5.1, where $\mu = 299.8$ ms, $\sigma = 26$ ms and $k = 0.29$.

$$f_{3G}(x \mid k, \mu, \sigma) = \frac{1}{\sigma} e^{-\left(1+k\left(\frac{x-\mu}{\sigma}\right)^{\frac{1}{k}}\right)} \left(1 + k\frac{x-\mu}{\sigma}\right)^{\frac{-1}{k}} \tag{5.1}$$

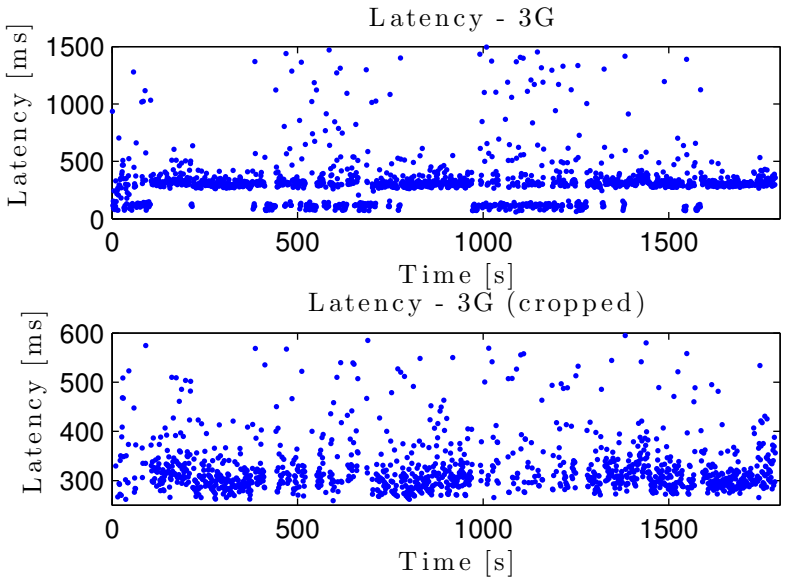

Figure 5.1 Latency plot for the 3G network. The y-axis describes the round-trip latency recorded for each of the packet sent. There was one packet sent every second. The upper plot shows the all the data collected and the lower figure shows the cropped latency data — where any latency above 600 ms or below 250 ms has been excluded.

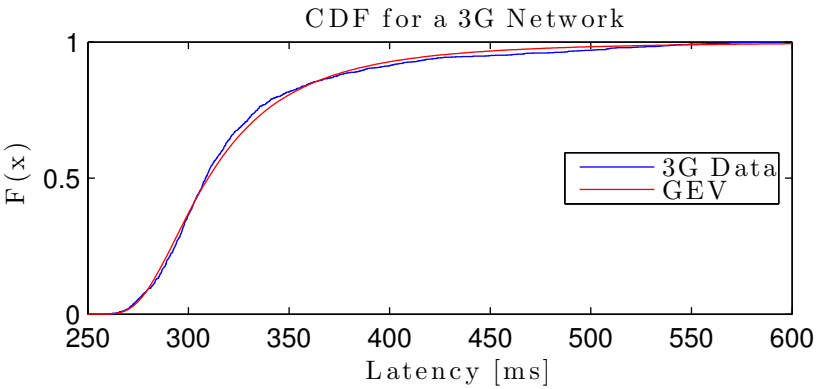

Figure 5.2 Cumulative distribution function for the 3G latency data fitted with a generalized extreme value distribution.

The results for the latency measurements of the 4G network can be seen in Fig. 5.3. As in the 3G latency plot the y-axis shows the round-trip delay for the packets. The result is indeed a very peculiar plot. Looking at the upper image there is something that can be described as a saw-tooth plot; zooming in on one of the bands, as shown in the lower image, there is yet another saw-tooth plot. The reason for this was unknown for us and could be one of many things. For instance it could depend on the OS of the computer used to run the tests, it could be the scheduler in one of the base stations or in the firmware of the Huawei dongles. Regardless, this was outside the scope of this thesis and not investigated any further; since it was only the mean latency that were of interest. The mean latency was concentrated around 60 ms with a maximum value around 65 ms and a minimum value around 50 ms.

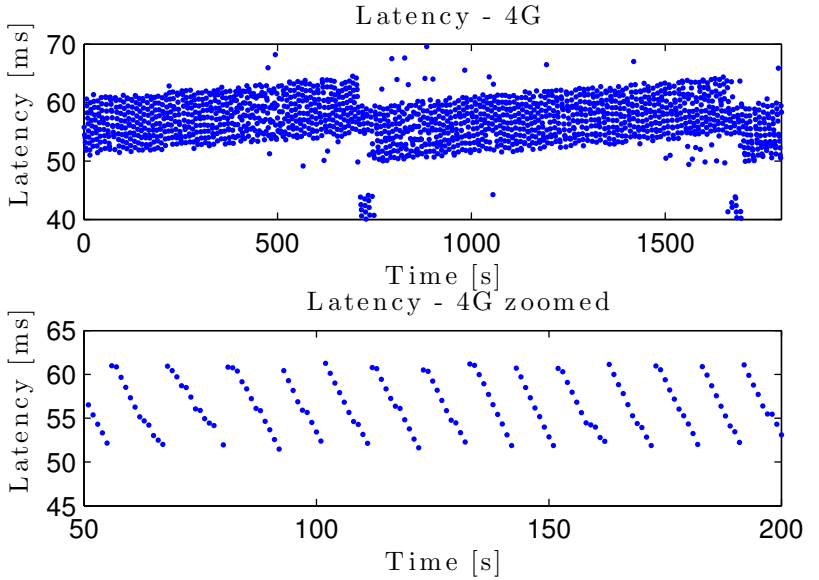

Figure 5.3 Latency plot for the 4G network. The y-axis describes the round-trip latency recorded for each of the packet sent. There was a packet sent once every second. The lower plot illustrates just how peculiar the data is.

As with the 3G network the 4G latency data was fitted with a distribution using the Matlab tool *'dfittool'*. The results was a Normal distribution. The cumulative distribution function (CDF) of the fitted distribution along with the 4G latency data can be seen in Fig. 5.4. The normal distribution can be expressed according to Eq. (5.2), with $\mu = 60$ ms and $\sigma = 5$ ms.

$$f_{4G}(x,\sigma,\mu) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{5.2}$$

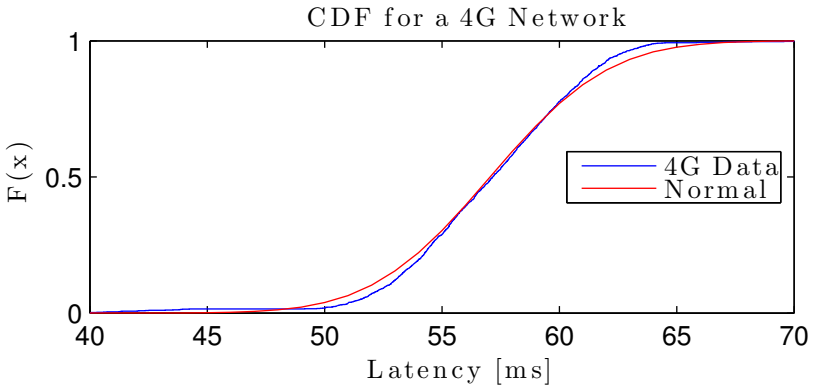

Figure 5.4 Cumulative distribution function of the 4G latency data along with the fitted normal distribution.

The latency for the wired network was assumed to have no latency, since it will be negligible in comparison with the one of the 4G and 3G networks. The 5G network was assumed to be somewhere in between that of the 4G and the wired network. Thus the 5G network was chosen to be modeled using Eq. (5.3), with $\mu = 10$ ms and $\sigma = 1$ ms.

$$f_{5G}(x,\sigma,\mu) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{5.3}$$

The summary of the distributions fitted, and their values, is shown in Table 5.1.

Network	Distribution	μ [ms]	σ [ms]	k
3G	GEV	299.77	25.96	0.292
4G	Normal	60	5	-
5G	Normal	20	2	1
Wired Network	n/a	0	0	-

Table 5.1 Table showing the models chosen for simulating latency over four different network types, Wired, 3G, 4G, and 5G.

5.2 Latency Simulation

The result of the simulated latency can be seen in Fig. 5.5. The different networks have been simulated in the following order: *Wired Network*, *5G Network*, *4G Network*, and *3G Network*. The wired network had no added latency, the 5G had about 10 ms one-way delay, the 4G had a one-way delay of about 30 ms and the 3G network had a one-way delay of about 150 ms.

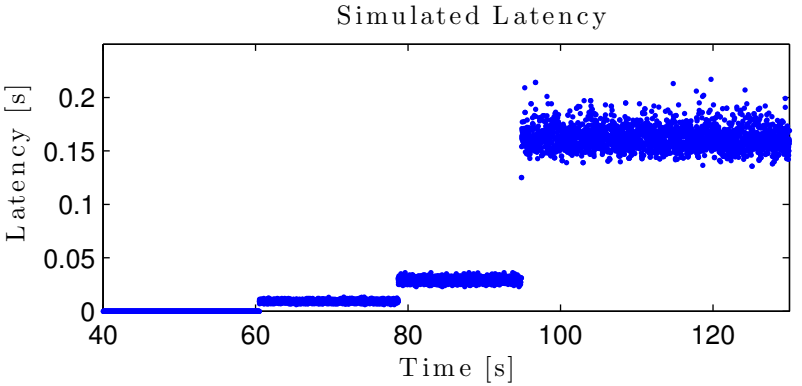

Figure 5.5 Network type and simulated latency during the collision test. First the wired network was simulated (no added latency), then the 5G network, the 4G network and last the 3G network was simulated.

5.3 Collision Test

Figure 5.6 shows the velocity and reaction forces measured at the Tool frame on the robot during a collision for each on the different network modes. It shows them in the x-direction, the direction of the collision. Each collision had a collision force of about 4 N. The upper left image shows the collision for the wired network, the upper right for the 5G network, the lower left for the 4G network, and the lower right image shows the collision for the 3G network.

Looking at the collisions in Fig. 5.6 one can see that there is a difference in behavior between the four network modes. For the wired network and the 5G network the reaction to the collision occurs in one smooth reaction. Note however that the tip is slightly smoother for the 5G network than for the wired network. There is a drawback motion moving the robot away from the collision surface. For the 4G network there is two distinct peaks during the reaction-phase to the collision. The first peak is due to the local controller at the robot-interface and occurs instantly after the collision. The second peak, about 60 ms, after the first peak is due to the reaction of the operator controlling the haptic interface. The delay in that peak happens

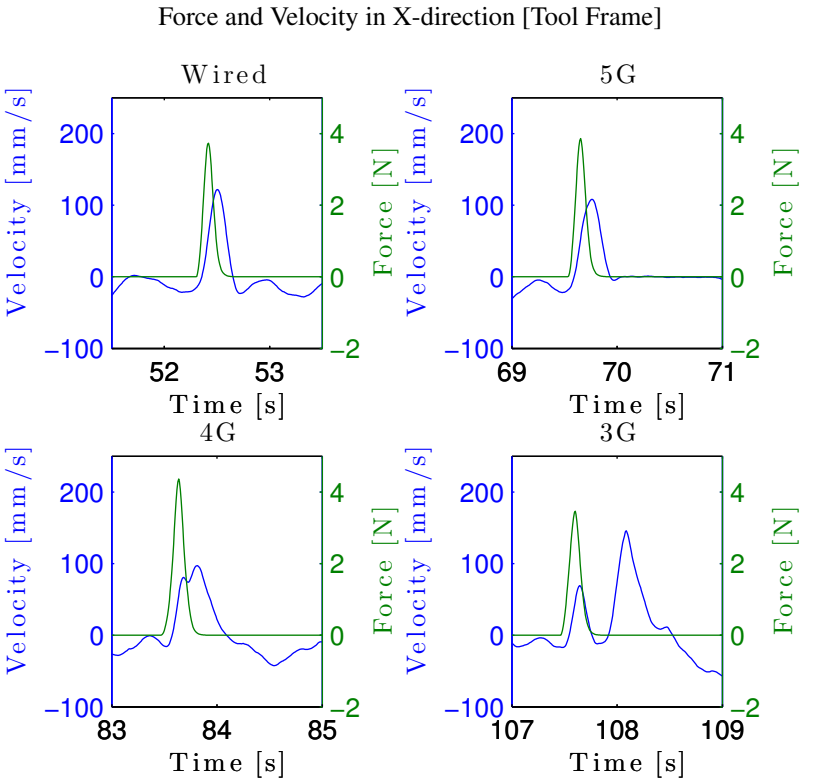

Figure 5.6 Collision test for the four network modes Wired network (upper left), 5G network (upper right), 4G network (lower left), and 3G network (lower right). The left y-axis show velocity, the right y-axis show contact force and the x-axis show the time-scale. The velocity and position are in x-direction in Tool frame coordinates.

because the signal has to travel from the robot to the haptic device, interact with the operator, and then travel back to the robot. For the 3G network there is no longer one reaction with two peaks, but rather two distinct reaction — one coming from the the robot controller and one from the operator. The time difference between the two reactions correlates with the round-trip delay of the 3G network, about 300 ms.

The same four collision shown in Fig. 5.6 are also illustrated in Fig. 5.7 but instead of showing the force-velocity relationship it shows the position-force relationship. The position and forces are shown in z-direction in the Task Frame — in order to show the absolute position of the collision. Comparing the four plots in Fig. 5.7 it is clear that there is a greater pullback reaction from the collision when running on a slower network. The wired and 5G network have a pullback of about 20 mm, the 4G network about 30 mm and the 3G network has a pullback reaction

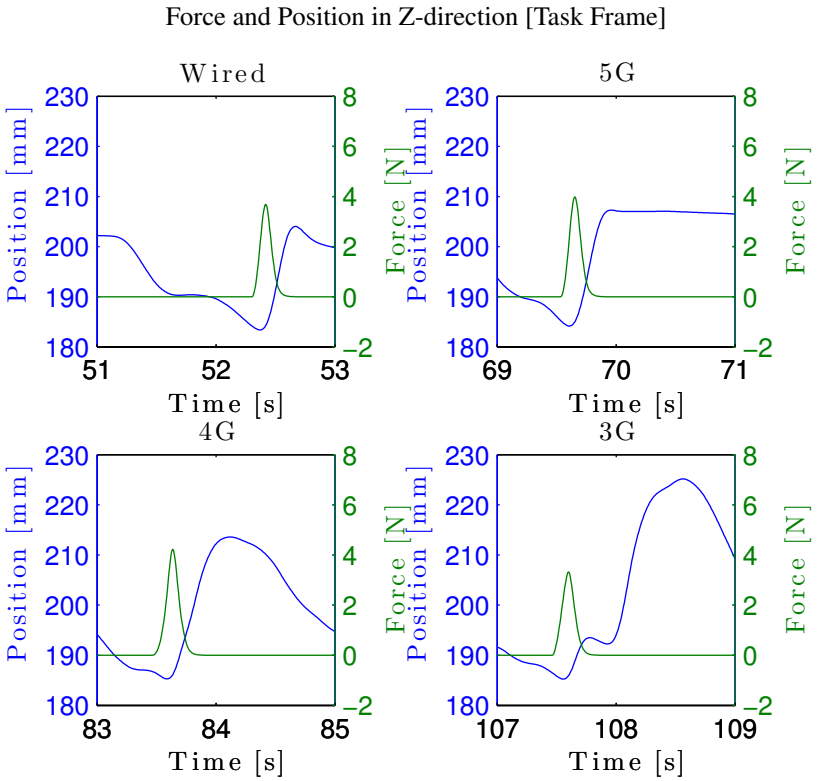

Figure 5.7 Collision test for the four network modes Wired network (upper left), 5G network (upper right), 4G network (lower left), and 3G network (lower right). The left y-axis show position in millimeter, the right y-axis show contact force and the x-axis show the time-scale. The position and forces are in the z-direction in Task frame coordinates.

of about 40 mm. In the 3G network plot there is also two very distinct pullback motions — one due to the internal controller in the robot interface and one, that is much greater, due to the reaction of the operator.

5.4 Dexterity Test

The result of the dexterity test is shown in Fig. 5.8. The y-axis shows the time it took to complete the test and the x-axis show the different network modes. The plot show the mean time to complete the test, green circles, along with its standard deviation, the blue bar. Clearly when performing the test on the wired network or the 5G and 4G network the test subject were quicker to finish the test than when running on

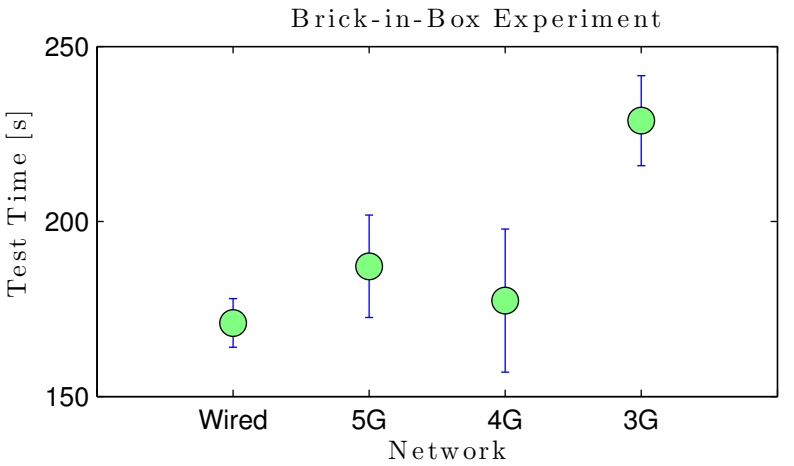

Figure 5.8 Results of the dexterity tests showing the average time left, as well the standard deviation, after putting the six bricks into the box.

the 3G network. Wired network is the fastest with the lowest standard deviation. Interestingly enough the 4G network is slightly faster than the 5G network but has a higher standard deviation. The reason for this is that the test subject "got lucky" and was able to find the hole in the box on the first try, thus obtaining super fast test time. In Table 5.2 all the test results from the different tests run on the different network modes can be seen.

Time to Finish Dexterity Test [s]

Test Run \ Network	Wired	5G	4G	3G
1	179	203	178	206
2	177	170	210	237
3	168	175	155	233
4	169	188	167	235
5	162	200	177	233
Mean	171	287.2	177.4	228.8
Standard Deviation	6.9	14.7	20.5	12.9

Table 5.2 Table over when the different network types were used during the collision test.

Stability Test

Figure 5.9 shows the results of these tests for the different network modes. As before the upper left one shows the result for the wired network, the upper right one shows

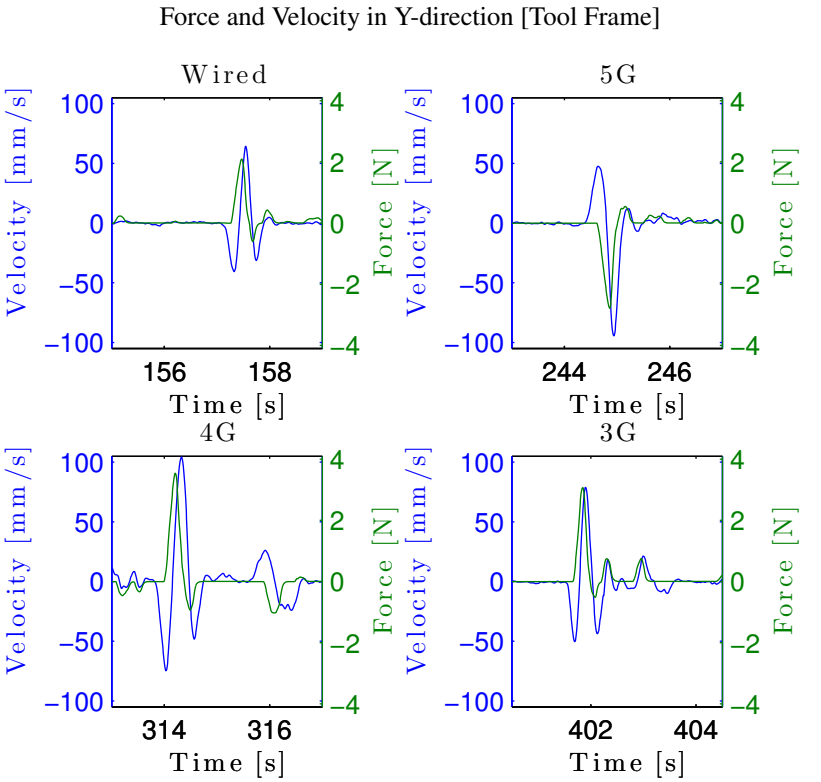

Figure 5.9 In-hole stability test for the four network modes Wired network (upper left), 5G network (upper right), 4G network (lower left), and 3G network (lower right). The left y-axis show velocity (mm/s), the right y-axis show contact force (N) and the x-axis show the time-scale (s). The velocity and position are in y-direction in Tool frame coordinates.

the 5G network, the lower left one shows the 4G network and the lower right one shows the result when running on the 3G network. The left y-axis shows the velocity (mm/s) in the y-direction, direction of collisions, the right y-axis shows the contact forces (N) and the x-axis shows the time (s). All is expressed in the Tool frame coordinate system. Clearly, all four systems are stable since they all decay to zero velocity. However the wired network, 5G network and 4G network decay faster than the 3G network. Furthermore the wired connection have a smaller reaction velocity than the others. The 5G network and the 4G network have about the same reaction velocity, peaking at about 100 mm/s. However, it is worth to mention that there is no major difference in the stability behavior when running on the four different network modes.

6

Discussion

It is very difficult to find good research on how the quality of a bilateral teleoperation can be expressed. It is even harder to find research about how latency affects the quality of a bilateral teleoperation. This is expressed in [Steinbach et al., 2012], when they say that an accelerated progress of haptic communications is hindered due to a lack of objective quality metrics. Up until now, most research have been focusing on the stability of a bilateral teleoperation in the presence of communication latency. [Aziminejad et al., 2006] discussed how they were using wave variables in order to achieve stability and transparency for a network latency up to 200 ms. However, they did not properly investigate how this affected the quality, and the operators ability to perform a task using their setup. Furthermore, [Tavakoli et al., 2007] tested whether visual feedback was better that haptic feedback when controlling a medical robot, and they concluded that haptic feedback was indeed better. In their work, they pressed that future work should focus on human studies to compare performance in the presence of communication latency.

In this work, we did come up with an approach to put an objective quality metric on the haptic-feedback application. Without such an approach, it would be very difficult to suggest any upper bound on what a future network latency should be in order to host such an application on it.

The conclusion, an upper bound of 20 ms in communication latency, is slightly higher than what is proposed in [Fettweis, 2014]. In their work, they investigated what it would take in order for humans to steer their surrounding real and virtual environment. They said that, if possible, it would give rise to the Tactile Internet. They concluded that a Tactile Internet would have to have a maximum round-trip latency of 1 ms.

6.1 What Could Be Better?

The biggest performance issues with the resulting setup used in this work is that of not being able to do force control. Although force control was not the main goal, it was definitely a highly desired goal. Not being able to do force control means not

being able to push things around in the environment. The two main causes, for this issue, are delay in the communication and the local robot controller.

Delay in the communication between the haptic interface and the robot causes the effects described in the section above, the accumulation of forces before being actuated to the operator. For a truly realistic and transparent teleoperation setup the operator should feel that the forces actuated on him is the result of his movements and not the other way around.

The local robot controller results in the dissipation of contact forces and thus making the system safer and more robust with regards to collisions, but it also diminish the possibility of force control.

6.2 Future Work

For future work I would suggest splitting the problem into two parts: studying the human mind with regards to motion control and building a virtual environment with the possibility of simulating different communication modes.

Studying the human mind and nervous system might result in new ways of structuring the bilateral teleoperation setup. It could also bring a deeper understanding to what makes a setup "realistic" in the eyes of the operator.

Building a virtual environment with focus on the possibility to simulate different types of delays as well as the to quickly build and try different controllers would bring great opportunities to further study the effects of delays in a very controlled environment. It would also offer the possibility to quickly build and try new types of controllers without the risk of damaging an expensive robot or the environment. Furthermore it could serve as a great platform to study what factors causes the bilateral teleoperation to be intuitive; since it would be very easy to allow test subjects to perform various tasks in the virtual environment.

A different path for future work would be to extend the work done in this thesis and work with the a two armed robot with force feedback, to develop a cool demonstration.

6.3 A General Discussion

The future of bilateral teleoperation is likely to develop in order to perform tasks highly demanding in dexterity and taking place in an environment unknown to the robot or the computer, thus needing control from a user. The greatest advantage will be the combination of a robots mobility, strength, precision, and resilience along with the operators great knowledge of the task to be perform and the intuition it brings. One example we see today are the surgical robots, although not yet with force feedback. Another example is challenge to build humanoid robots that interact and move around in an environment after a disaster. I believe it will be possible to control certain parts of such robots through bilateral teleoperation when necessary.

For instance, imagine if the robot would have to solve a task that it has not been programmed for, a good intuitive bilateral teloperation setup could provide a quick and good solution.

Changing the view to the future of Cloud computing and the Internet of Things i believe it will take quite a while before we are there. Some predict in 2020 but I think it is reasonable to say it take somewhat longer. The reason being the security issues they bring. Many of the major companies around the globe today have very strict security standards rendering them unable to adopt the Cloud even if it would be readily accessible for them. I think it is safe to say that it will not be them who drive the evolution of the IoT and Cloud, but rather the small, new start-up companies from all across the world. However, if they are able to resolve the security issue that could lead to having a more secure storage of data in the Cloud than what is now available I believe we will see an explosion in Cloud computing and IoT.

7

Conclusion

The result of the dexterity test, Fig. 5.8, shows that the system is good enough to complete the experiments regardless of which network is simulated. It shows that there is a difference between the 3G network and the rest, but no significant difference between the wired network, 5G network nor the 4G network.

Furthermore if one looks at the results in Figs. 5.6 and 5.7, from the collision test, there is no change in behavior between the wired network and the 5G network. However, when going from the 5G network to the 4G network there is subtle change in behavior. In the 4G network there is two distinct velocity peaks, seen in Fig. 5.6. The reason being that the round-trip delay becomes big enough to start playing a role, if just a minor one. When changing from the 4G network to the even slower 3G network the changes seen before becomes exaggerated. Looking at the velocity-force plot there is now two distinct velocity curves, rather than just two peaks in one, and looking at the position-force plot there is clear evidence of two different pullback motions. The first one is due to the local robot controller and the other, about 300 ms after, is due to the operator reaction.

It is worth noting that the slightly different behavior seen when running on the 4G network did not affect the quality of the bilateral teleoperation to any significant length. The standard deviation of the total test time for 4G was larger than that of the wired- and the 5G network, but the mean time was not significantly different.

Thus, it is reasonable to conclude that the latency of the 4G network, a round-trip delay of 60 ms, provides an upper boundary of what is desirable when running this application on a network mode. A goal for the round-trip delay would be that of the 5G network, about 20 ms, simulated in this experiment, the reason being that it shows no difference in behavior, when looking in depth for one, to that of the wired network.

Bibliography

AmazonWebServices (2013). *Overview of Amazon Web Services*. visited on Aug 2014. URL: http://media.amazonwebservices.com/AWS_Overview.pdf.

— (2014). *AWS case study: Pfizer*. visited on Aug 2014. URL: http://aws.amazon.com/solutions/case-studies/pfizer/.

Armbrust, M., A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia (2009). *Above the Clouds: A Berkeley View of Cloud Computing*. Tech. rep. UCB/EECS-2009-28. EECS Department, University of California, Berkeley.

Aziminejad, A, M. Moallem, and R. Patel (2006). "A study of transparency in wave-based teleoperation". In: *Mechatronics, 2006 IEEE International Conference on*, pp. 392–397.

Blomdell, A., G. Bolmsjö, T. Brogårdh, P. Cederberg, M. Isaksson, R. Johansson, M. Haage, K. Nilsson, M. Olsson, T. Olsson, A. Robertsson, and J. Wang (2005). "Extending an industrial robot controller–Implementation and applications of a fast open sensor interface". *IEEE Robotics & Automation Magazine* **12**:3, pp. 85–94.

Blomdell, A., I. Dressler, K. Nilsson, and A. Robertsson (2010). "Flexible application development and high-performance motion control based on external sensing and reconfiguration of ABB industrial robot controllers". In: *2010 IEEE International Conference on Robotics and Automation*. Anchorage, Alaska.

Buyya, R., C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic (2009). "Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility". *Future Generation Computer Systems* **25**:6, pp. 599 –616. ISSN: 0167-739X.

Capek, K. and D. Wyllie (2010). *R.U.R (Rossum's Universal Robots)*. Echo Library. ISBN: 9781406867114.

Cardwell, D. (2014). *At Newark airport, the lights are on, and they're watching you*. visited on Aug 2014. URL: http://www.nytimes.com/2014/02/18/business/at-newark-airport-the-lights-are-on-and-theyre-watching-you.html?_r=0.

Cisco (2013a). *Cisco global Cloud index: forecast and methodology, 2012-2017*. visited on Aug 2014. URL: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf.

— (2013b). *Fog Computing, ecosystem, architecture and applications*. visited on Aug 2014. URL: http://www.cisco.com/web/about/ac50/ac207/crc_new/university/RFP/rfp13078.html.

Coetzee, L. and J. Eksteen (2011). "The Internet of Things - promise for the future? an introduction". In: *IST-Africa Conference Proceedings, 2011*, pp. 1–9.

Corke, P. (2011). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Tracts in Advanced Robotics. Springer. ISBN: 9783642201431.

Dignan, L. (2013). *Internet of Things: $8.9 trillion market in 2020, 212 billion connected things*. visited on Aug 2014. URL: http://www.zdnet.com/internet-of-things-8-9-trillion-market-in-2020-212-billion-connected-things-7000021516/.

Ericsson (2013). *5G Radio Access*. visited on Aug 2014. URL: http://www.ericsson.com/res/docs/whitepapers/wp-5g.pdf.

Evans, P. C. and M. Annunziata (2012). *Industrial Internet: pushing the boundaries of minds and machines*. visited on Aug 2014. URL: http://files.gereports.com/wp-content/uploads/2012/11/ge-industrial-internet-vision-paper.pdf.

Fettweis, G. (2014). "The Tactile Internet: applications and challenges". *Vehicular Technology Magazine, IEEE* **9**:1, pp. 64–70. ISSN: 1556-6072.

Force Dimension (2014). *Omega.7 specsheet*. visited on Aug 2014. URL: http://www.forcedimension.com/downloads/specs/specsheet-omega.7.pdf.

Garcia, E., M. Jimenez, P. De Santos, and M. Armada (2007). "The evolution of robotics research". *Robotics Automation Magazine, IEEE* **14**:1, pp. 90–103. ISSN: 1070-9932. DOI: 10.1109/MRA.2007.339608.

Gartner IT Glossary (2014). *Cloud Computing definition*. visited on Aug 2014. URL: http://www.gartner.com/it-glossary/cloud-computing.

Harbor Research (2014). *Are you prepared for big changes in the way we will learn, work and innovate?* visited on Aug 2014. URL: http://harborresearch.com/are-you-prepared-for-big-changes-in-the-way-we-will-learn-work-and-innovate/.

Hartenberg, R. S. and J. Denavit (1964). *Kinematic Synthesis of Linkages*. McGraw-Hill, New York.

Hokayem, P. F. and M. W. Spong (2006). "Bilateral teleoperation: an historical survey". *Automatica* **42**:12, pp. 2035 –2057. ISSN: 0005-1098.

IBM (2014a). *Smarter care at MD Anderson*. visited on Aug 2014. URL: http://www-03.ibm.com/software/businesscasestudies/us/en/corp?synkey=Q531267D50817F22.

IBM (2014b). *What is Watson?* visited on Aug 2014. URL: http://www.ibm.com/smarterplanet/us/en/ibmwatson/.

IFTT (2014). *What is IFTTT?* visited on Aug 2014. URL: https://ifttt.com/wtf.

Malik, O. (2013). *Amazon CTO Werver Vogels: Cloud and SaaS are going global, fast*. visited on Aug 2014. URL: http://gigaom.com/2013/12/30/amazon-cto-werner-vogels-cloud-and-saas-are-going-global-fast/.

Munjin, D. and J. Morin (2012). "Toward Internet of Things application markets". In: *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, pp. 156–162.

Niemeyer, G. and J.-J. Slotine (1991). "Stable adaptive teleoperation". *Oceanic Engineering, IEEE Journal of* **16**:1, pp. 152–162. ISSN: 0364-9059.

Pretz, K. (2013). *The next evolution of the Internet*. visited on Aug 2014. URL: http://theinstitute.ieee.org/technology-focus/technology-topic/the-next-evolution-of-the-internet.

Spong, M., S. Hutchinson, and M. Vidyasagar (2005). *Robot Modeling and Control*. Wiley. ISBN: 9780471649908.

Steinbach, E., S. Hirche, M. Ernst, F. Brandi, R. Chaudhari, J. Kammerl, and I Vittorias (2012). "Haptic communications". *Proceedings of the IEEE* **100**:4, pp. 937–956. ISSN: 0018-9219.

Tan, L. and N. Wang (2010). "Future Internet: the Internet of Things". In: *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*. Vol. 5, pp. V5–376–V5–380.

Tavakoli, M., A Aziminejad, R. Patel, and M. Moallem (2007). "High-Fidelity bilateral teleoperation systems and the effect of multimodal haptics". *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **37**:6, pp. 1512–1528. ISSN: 1083-4419.

Texas Instruments (2013). *The evolution of the Internet of Things*. visited on Aug 2014. URL: http://www.ti.com/lit/ml/swrb028/swrb028.pdf.

Vance, A. (2014). *Amazon's Cloud is one of the fastest-growing software businesses in history*. visited on Aug 2014. URL: http://www.businessweek.com/articles/2014-07-15/amazons-cloud-is-one-of-the-fastest-growing-software-businesses-in-history.

William Boldt, A. (2014). *The evolution and DNA of IoT*. visited on Aug 2014. URL: http://atmelcorporation.wordpress.com/2014/07/29/ioxt/.

Xenomai (2014). visited on Aug 2014. URL: http://xenomai.org/start-here/.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER´S THESIS

Date of issue
August 2014

Document Number
ISRN LUTFD2/TFRT--5951--SE

Author(s)
Victor Millnert

Supervisor
Johan Eker, Ericsson R&D, Dept. of Automatic Control, Lund University, Sweden
Mahdi Ghazaei, Dept. of Automatic Control, Lund University, Sweden
Anders Robertsson, Dept. of Automatic Control, Lund University, Sweden
Rolf Johansson, Dept. of Automatic Control, Lund University, Sweden (examiner)

Sponsoring organization

Title and subtitle
Quality-Latency Trade-Off in Bilateral Teleoperation

Abstract

The purpose of this thesis is to investigate how the latency in mobile networks affects the quality of highly demanding and sensitive applications running on it. Furthermore, this thesis will provide some information to what is going on in the field of Cloud Computing and the Internet of Things. It will hopefully spark a discussion about what possibilities will come with the development of the Cloud and Internet of Things.

The application chosen was a bilateral teleoperation, with force feedback, controlled in 6 dimensions. To investigate how the quality depends on network latency, different network models were simulated as the communication channel. The networks chosen to be simulated were a 3G, 4G, and a 5G cellular network along with a wired network chosen as a baseline.

On this setup two main experiments were done. The first one was a collision test and the second one a dexterity test, where a user was supposed to pick up a small wooden brick and put it into a box. The results from the experiments showed that there was indeed a difference in behavior when having a network delay larger than 20 ms.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-61

Recipient's notes

Security classification