University of Twente

EEMCS/Electrical Engineering
Control Engineering

Implementation of a 2D Master-Slave system

Bayan Babakhani

MSc Report

Supervisors:
prof.dr.ir. S. Stramigioli
prof.dr.ir. J. van Amerongen
M.C.J. Franken MSc

April 2008

Report nr. 006CE2008
Control Engineering
EE-Math-CS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

A teleoperator is a device used by humans to remotely interact with either a virtual environment or another device. The device that is used to define the desired task is called the master and the one performing these tasks is called the slave. Force feedback to the user of a teleoperator can considerably improve the ability to perform complex tasks that interact with the environment. The technology that interfaces an environment to the user using the sense of touch and thus provides force feedback is called haptic technology.

The stability of such devices is very crucial as the lack of it can cause unnatural feedback to the operator or even damage the operator and/or the environment. The stability analysis of a haptic device is very challenging due to the complexity and the non-linearity of the environment and dynamics of the human operator. A simple but powerful method to ensure stability is the passivity approach. Passivity is a sufficient condition for stability. Passivity examination can be done by monitoring the energy of a system. When looking at a system from the energetic point of view, energy leakage or production can lead to unstable behaviour.

During this project a demonstrator for haptic feedback principles has been realized. The Pantograph Mk-II [6] is used for this purpose. The two set-ups in this system both function as master and slave at the same time. On this master-slave system, theories that guarantee passivity can be implemented.
An intrinsically passive controller has been implemented that operates in an energy consistent fashion. Simulation and tests have proven that this controller remains stable independent of the sample rate. Furthermore, the scattering formalism has been implemented to keep the haptic system stable despite variable communication delays and loss of data. The system is shown to sustain its stability despite the unreliability of the communication between the devices and low sample frequency.

Concluding, this report elaborates on the underlying theory that is used in this project for guaranteeing passivity and thus stability. Next to the theoretical background, the implementation of an intrinsically passive controller and wave scattering transformation is discussed.

Preface

This is the end of a very educating period of my life, which would not have been as rewarding and fun without the people who supported and inspired me. I would like to use this opportunity to express my thanks to some of them.

I would like to thank the following people for their valuable input during my master project;
My supervisors throughout the project, prof. van Amerongen, prof. Stramigioli, Gijs van Oort, Edwin Dertien and Michel Franken, for good advices and technical support.
The technical staff at CE; Gerben te Riet o/g Scholten and Marcel Schwirtz for providing me with tools and software and Alfred de Vries for putting the used set-ups together.
My thanks also go to Marcel Groothuis for his hardware and modelling related support, Frank Groen for his support with 20-sim and Peter Visser for his support with 4C tool chain.
Also Carla Gouw-Banse is thanked for squeezing me into crowded agendas.

The CE staff and students are thanked for pleasant talks during the coffee brakes. Jos, Jan and Ludo are specially thanked for helping me with software problems and for giving useful hints. Also Leon Abelmann is thanked for his effort, support and good advices during my bachelor study.

And last but certainly not least, I would like to thank my family and friends for their support. My special thanks go to my mum, Rooni and Bert for putting up with me lately when I was moody and cheering me up. Thanks BaZiba for your endless love and encouragement, Rooni for your enthusiasm and mental support and Bert for comforting me and the extensive technical support that was lifesaving at times ☺.

Bayan
Enschede, April 2008

Abbreviations

b	damper constant
D	dimensional
DOF	degrees of freedom
e	effort
E	energy
f	flow
H	Hamiltonian
IPC	intrinsic passive controller
k	spring constant
p_w	penetration depth in the wall
$\dot{p_w}$	penetration velocity
P	power
PWM	pulse width modulation
q	position
Δq	displacement
T	sample time
T_d	delay time
VE	virtual environment
x	state
Z	impedance
ZOH	zero order hold

Contents

Chapter 1

Introduction

Humans use robots to do tasks that are for instance repetitive or in some sense difficult to perform. An example of a difficult task is performing an action in hard to reach or even unreachable places. For this purpose, a teleoperator can be used. A teleoperator is a device used by humans to remotely interact with either a virtual environment or another device in a certain environment. The device that is used to define the desired task is called the master and the device that performs tasks ordered by the master is called the slave.
Control of a teleoperator is usually done visually. This means that the operator decides on his action by looking at the current state of the teleoperator. Visual feedback is however insufficient when dealing with a delicate environment or when performing complicated tasks. Force feedback to the user of a teleoperator can considerably improve the ability to perform complex tasks that interact with the environment. A teleoperator is said to be controlled bilaterally, when force and/or tactile information are reflected by the slave to the master (and thus the operator).
Haptic technology refers to technology which interfaces an environment to the user using the

(a) MicroScribe 3D digitizer [11]

(b) 3D modelling using the Phantom [7]

(c) Haptic glove [5]

Figure 1.1: Haptic devices

sense of touch. Haptics is for instance useful in virtual reality and computer games to obtain a better feeling of reality, but is also very convenient, illustrative and time saving for 3D modelling and design (see Figure 1.1). Haptic devices can also be used for interfacing visually challenged people with computers. Another application can be found in the field of education; flight and space simulators are safer and relatively cheaper alternatives for educating purposes.
Haptics also has a considerable potential in the medical world. Next to training, this technology can be used in minimal invasive surgery. This kind of surgery enables surgeons to operate on patients with less trauma, pain and scarring. The operation is thus less intrusive and that results in faster recovery of the patient. Figure 1.2 shows the 'da Vinci' robot that can be used for minimal invasive surgeries. The surgeon can look inside the patient by means of a camera and operates remotely using two joysticks that control the arms inserted in the patient. Haptic feedback can contribute to the improvement of this system by letting the surgeon feel the environment in which he is

Figure 1.2: Da Vinci; the robot used for minimal invasive surgery [10]

operating. This decreases the chance of accidentally damaging tissues and organs, and provides the doctor with more information by enabling him to use his sense of touch.

These examples make the importance of stability of such devices very clear. Unstable behaviour in haptic systems can cause unnatural feedback to the operator or even damage the operator or the environment the system operates in. When looking at a system from the energetic point of view, unstable behaviour can be caused by energy leakage or energy production. There are two points in a discrete teleoperator where energy leakage or production can occur; the communication channel between the end-effectors of a teleoperator and the Zero Order Hold connecting the digital system (controller and/or virtual environment) to the continuous world.

When power variables are transmitted through the communication channel, the power at each side is equal to the product of the power variables of the other side. Consider a bilateral master-slave system with a communication delay of δ. This means that

$$\begin{aligned} f_{slave}(t) &= f_{master}(t-\delta) \\ e_{master}(t) &= e_{slave}(t-\delta) \end{aligned} \tag{1.1}$$

So for instance the power delivered at the slave side, for instance at time t is equal to

$$P_{slave}(t) = e_{slave}^T(t) f_{slave}(t) = e_{master}^T(t+\delta) f_{master}(t-\delta) \tag{1.2}$$

Thus the power is the product of the flow and the effort that refer to different time instants. This leads to power inconsistency, which can result in energy production [23].

ZOH keeps a power variable constant during a sample period. The value of its conjugate variable can however change in the meantime, such that extra energy could be introduced to the system [28]. An example of a situation in which ZOH contributes to energy production is when a haptic device is in contact with a rigid environment like a (virtual) wall. When the user withdraws the haptic device from the wall, the discrete controller keeps its previous output until the next sample moment. This means that the force produced by the haptic interface is larger than what it is supposed to be. Thus the operator receives energy from the haptic interface, which may cause instability.

There are many definitions of stability in system theory and thus different ways to analyse the stability of a system. A few examples are BIBO stability (bounded-input bounded output)and Lyapunov stability. However, the non-linear environment and dynamics of the human operator are very challenging to characterize and thus complex to model. This makes the stability analysis of a haptic device very difficult. A simple but powerful method to ensure stability is the passivity approach. Passivity is a sufficient condition for stability. Passivity formulation states that the combination of two passive systems, connected either in a feedback or parallel configuration, is

again passive. The stored energy and power dissipation of the combined system is obtained by adding the individual functions of both systems [21].

1.1 The assignment

The goal of this assignment is to build a demonstrator for haptic feedback principles. To this end, a master-slave system should be realized. The set-ups used, should be identical to each other, not only in the hardware sense, but also in their status. This means that both set-ups are master to each other and, at the same time, function as a slave that should follow the other set-up's reference. On this master-slave system, the principles that guarantee passivity should be implemented. Both the unreliable channel and the ZOH that are the bottle-necks of the system, are to be implemented in such a way that the system remains stable, despite the discrete nature of the system and despite the loss and delay of data in the communication between the master and slave.
The focus of this assignment is thus on guaranteeing stability in a haptic bilateral telemanipulator. Hence, a stable discrete control and a stable communication should be implemented. This means that other aspects of this project such as a good haptic feeling and design and/or optimization of the system from the electronic and mechanical perspective are chosen to be left out of the scope of this assignment .

1.2 Report overview

This report starts with the theoretical background of the chosen approach in chapter 2. Chapter 3 expands on the implemented algorithms that are explained in chapter 2. The performance of the system is shown in this chapter by means of measurement results. The conclusions are presented in chapter 4 followed by recommendations for future work. This report concludes with appendices containing more detailed information on various parts of the project.

Chapter 2

Stable haptic bilateral telemanipulator

In haptic interaction, instability can occur due to the dynamics of (virtual) environments and the nature of the sampled data. As explained in chapter 1, passivity is a powerful tool to guarantee stability of a haptic system. Stability of a system, however, does not imply a good performance of that system. An apparently passive impedance is a sufficient condition for stability of an actively controlled system in contact with a passive environment [9]. [15] shows that the impedance of the human arm is highly adaptable and is carefully tuned to preserve stability in conditions of interest for haptic displays. This ensures the stability of the part of interaction concerning the human operator and the environment. Thus unstable behaviour in a haptic teleoperator, when interacting with the environment, can be caused either by time delays in the communication channel between the end-effectors of the system, or energy production due to the discrete nature of the used controllers.

2.1 Previous work

In current force-reflecting teleoperation applications, destabilisation due to delay is often dealt with by adding large amounts of damping at various locations throughout the system. This method not only does not always provide a formal stability guarantee, but it also can considerably limit the system performance and, furthermore, is typically very sensitive to the parameters describing the system. For instance adding damping to the communication channel is actually stabilising the standard communication by dissipating all the produced energy. This leads to a passive communication. However, when forces are reflected, the velocity is modified. Due to additional dissipating elements, the telemanipulator tracks the masters velocity, but not its position [1].
In [8], passivity is used to derive fixed parameter virtual coupling between a virtual environment and the haptic interface. This over conservative method can lead to poor performance due to high dissipation by the fixed damping value that is used to guarantee passivity under all operating conditions.
[14] uses a passivity observer (PO) and a passivity controller (PC) to ensure stable contacts under a wide variety of operating conditions. This method measures active system behaviour(PO) and injects variable damping (PC) whenever net energy is produced by the virtual environment [16].
The reader is referred to [17] for more control strategies for stability robustness.

A different passivity approach to guarantee the stability of a bilateral haptic teleoperator is using wave scattering formulation in combination with an Intrinsically Passive Controller (IPC). In order to achieve passivity of the communication channel and eliminate the effect of possible

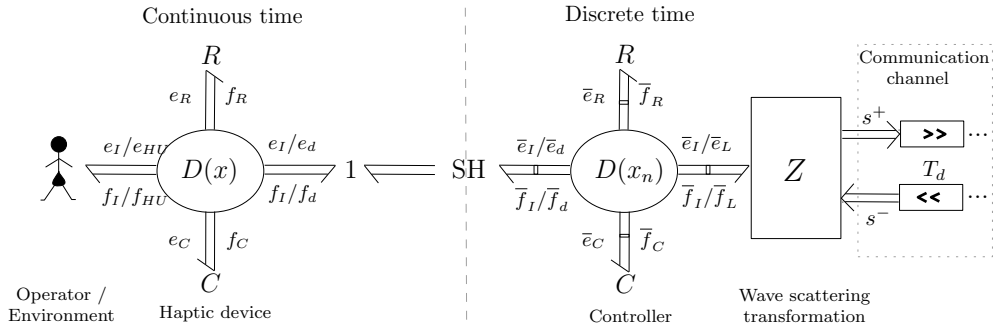


Figure 2.1: Total system

delays in the computation of the output, scattering theory can be used [24]. To guarantee the passivity of the controller of the haptic devices, an IPC can be implemented. This combination allows the manipulation of an arbitrary stable environment without loss of global stability. In this chapter the theoretical background of this approach is discussed. Section 2.2 elaborates on the port-Hamiltonian formalism, which is used in section 2.3 to explain the idea behind energy based control. Finally, section 2.4 illustrates the wave scattering transformation. Figure 2.1 shows the total system. This figure gives a complete and detailed overview of the implemented theories that are covered in the following sections.

2.2 Port-Hamiltonian systems

In a passive and thus stable system, the total energy should stay constant or decrease. To monitor the energy of a system, port-Hamiltonian formalism can be used. The port-Hamiltonian formalism is a modelling framework representing the physics of the system. Compared to other modelling frameworks like the behavioural framework or block diagrams, the port-Hamiltonian implementation provides most insight when the physics of the system are investigated since its structure directly shows energy flows inside the system [13].

A port Hamiltonian system is considered to be composed of [28]:

- A state manifold X
- An Energy function $H : X \rightarrow R$
- A network structure $D(x) = -D(x)^T$ whose graph has the mathematical structure of a Dirac structure
- An interconnection port represented by an effort-flow pair $(e, f) \in V^* \times V$

A discrete port-Hamiltonian system can be seen as a continuous time port-Hamiltonian system in which the power flow through each port is frozen during each sample period T. At each sample moment, these values can be adjusted such that an exact match between the energy of the discrete system and its continuous counterpart exists.

Consider a system composed of a discrete and a continuous part, in a stable environment, as a group of port Hamiltonian systems with ports through which energy exchange can take place. The interaction port is split up in three sub-ports, modelling different power flows. The port through which the system interacts with the (passive or active) environment is indicated by I. C is used for the port associated with the storage of energy. The dissipative part is represented by R [27].

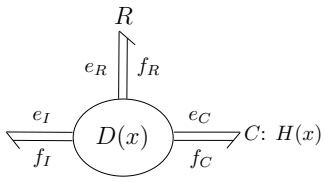


Figure 2.2: Interconnection subports of a port-Hamiltonian system

The system is passive if the total power supplied through the interaction power port, $P = -e_I^T f_I$, is equal to the time derivative of the stored energy E plus the power that is dissipated;

$$P = e^T f = \frac{dE}{dt} + P_{diss} \tag{2.1}$$

and

$$P_I + P_C + P_R := e_I^T f_I + e_C^T f_C + e_R^T f_R \tag{2.2}$$

which should always be equal to zero for a passive system, since no energy production should take place.

2.3 IPC

Intrinsic Passive Controllers belong to a class of controllers build of passive elements that do not produce any energy. An IPC has two ports; one connected to the system to be controlled and the other one to a supervisor module (higher level control). When the supervisor does not supply any power, the controlled system will be passive [27]. Errors in the plant model can never influence the passivity of an IPC. Thus, a system controlled by an IPC will stay stable when interacting with an (unknown) passive environment. The basic idea is to only use the energy flow into the system, through the continuous part, to control the plant such that the total energy stored in the discrete system does not exceed the total energy flow into the system. If the environment acts passive, the total energy injected into the discrete system is bounded over a limited time period and therefore the system will remain stable.

By using the energy balance at each sample moment, the total system can be kept stable due to the power consistency of the coupled system [23]. The supplied energy is equal to the energy flow through the interconnection port I minus the dissipated energy. The supplied energy through port I is equal to

$$\begin{aligned}\Delta H_I(n+1) &= \int_{nT}^{(n+1)T} e_d^T(n+1) f(s) ds = e_d^T(n+1) \int_{nT}^{(n+1)T} f(s) ds = e_d^T(n+1) T f_d(n+1) = \\ &= e_d^T(n+1)(q((n+1)T) - q(nT)) = e_d^T(n+1)\Delta q(n+1)\end{aligned} \tag{2.3}$$

where

$$f_d(n+1) = \frac{q((n+1)T) - q(nT)}{T} \tag{2.4}$$

$e((n+1)T)$ at the sample moment is equal to $e(nT)$ and $\Delta q(n+1)$, indicating the position change of the plant, is known. The part of ΔH_I that is dissipated by the damper can be calculated by

$$\Delta H_R(n+1) = \int_{nT}^{(n+1)T} e_R^T(n+1) f_R(n+1) ds = T f_R^T(n+1) R f_R(n+1) \tag{2.5}$$

hence, the change in the stored energy becomes

$$\Delta H_C(n) = -\Delta H_I(n) - \Delta H_R(n) = \Delta H(n) \tag{2.6}$$

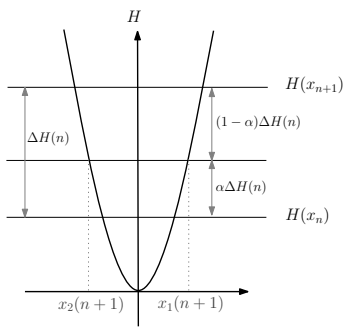


Figure 2.3: Choosing the next state based on the system energy

In order to preserve passivity, the state of the discrete system should have an energy level equal to

$$H(x(n+1)) = H(x(n)) + \Delta H(n+1) \tag{2.7}$$

which is basically the energy of the system one sample ago plus the energy change during the last sample period.
There are usually multiple states on the manifold that correspond to the same energy level greater than zero. The state closest to the current state should be chosen, because then the smallest jump is made to the next energy level. A zero energy level naturally corresponds to the state zero, in which the controller has no internal energy.

Energy Error/discrepancy

Due to changes in the environment during a sample period, it is possible that the energy of the system drops below zero; $H(x(n+1)) < 0$. Because of the discrete nature of the system, this effect is not detected before the next sample moment. This means that the system has already delivered more energy than it actually had. Two different alternatives for dealing with this situation will be explored here.
Since there is no state with negative energy level, the state with the minimum energy can be chosen; $x(n+1) = x_{min} = 0$. Lets call this alternative *minimum state*. This will cause dynamic deadlock since it prohibits supply of energy through port I by setting e_I equal to zero.
Another alternative is the so-called *energy leap*. This method chooses a state $x(n+1)$ with the same energy level as the current state $x(n)$. By doing so, the deadlock effect is avoided; by changing the sign of the controller output, the system dynamics are manipulated to absorb energy. The increase of internal energy can then be used to compensate for the energy discrepancy [27].

It is clear that in both cases an energy error bookkeeping should be used to maintain passivity of the discrete system over a limited time span, longer than one sample period. This bookkeeping should keep track of energy errors and the applied compensations.
Consider a discrete system consisting of a linear spring with state x having the quadratic energy function $H(n) = \frac{1}{2}kx_n^2$. Imagine a situation in which this system should loose a certain amount of energy, $\Delta H(n+1)$, greater than the system energy $H(n)$ in order to get to the energy level $H(n+1)$. Thus

$$|\Delta H(n+1)| > H(n) \quad : \quad \begin{cases} \text{mimimum state} & : \quad H_{error} \quad = \quad \Delta H(n+1) + \frac{1}{2}kx(n)^2 \\ \text{energy leap} & : \quad H_{error} \quad = \quad \Delta H(n+1) \end{cases} \tag{2.8}$$

The energy level of this system would change by $\frac{1}{2}kx(n)^2$ when choosing $x_{min} = 0$ as the next state. This leaves the system with an error equal to $H(n+1) + \frac{1}{2}kx(n)^2$. Using the second alternative, namely energy leap, the system keeps the same energy level, which means that the change

of energy is in some way 'ignored'. This would lead to an error equal to $\Delta H(n+1)$.

A positive $\Delta H(n+1)$ means that energy from the environment has been flowing into the system. When this occurs, the incoming energy can be (partially) used to compensate for the energy that once was used but was not actually available. The energy error will then decrease by $\alpha\Delta H(n+1)$ as shown in Figure 2.3, with $\alpha \in]0..1]$. The closer α is to 1, the faster the system recovers its passivity. On the other hand, a small α would have less influence on the dynamics of the system.

Deadlock effect

As mentioned before, when the state of the controller reaches its minimum value, $x = x_{min} = 0$, dynamic deadlock occurs. This deadlock is caused by the fact that the output force of the controller becomes zero, which makes P_I zero no matter the value of Δq.

$$\begin{cases} x &= 0 \\ e_d &= -kx \\ \Delta H_I &= e_d^T \Delta q \\ P_I &= \partial H/\partial t \end{cases} \quad \Rightarrow \; P_I = 0, \qquad \forall \Delta q \tag{2.9}$$

As an exception, $x(n+1)$ should be set to a carefully chosen value that is not based on the energy balance, but only on the knowledge that some energy exchange between the operator/environment and the plant has taken place. This energy exchange at the plant side is expressed in the form of the plant changing its position, which makes q in particular suitable for determining the new state of the system. The change of the controller state based only on the displacement of the plant is applied to resolve deadlock. Since it is not based on energy exchange at the interaction port of the controller, the energy introduced to the discrete controller as a result, should be compensated for in a later stage.

Inter-sample passivity loss

It should be noted that passivity loss between two samples is possible; for instance when during a sample period a considerable amount of energy flows into the system, but is withdrawn from it before the next sample in taken. After this sample period the net energy flow to the plant will be zero, but during the same period, there would be one moment at which the controller provided more energy than it had. This loss of passivity can never be noticed by the controller since it happens between the sample moments [27]. This is unavoidable because of the intrinsic loss of information due to the discrete nature of the system. But the important thing is that at each sample moment there is an exact match between the energy of the discrete controller and the continuous world.

2.4 Wave scattering

A wave transform encodes, without loss of information, the normal power variables of flow and effort, velocity and force, into wave variables, s^+ and s^- (see Figure 2.4). Wave variables represent

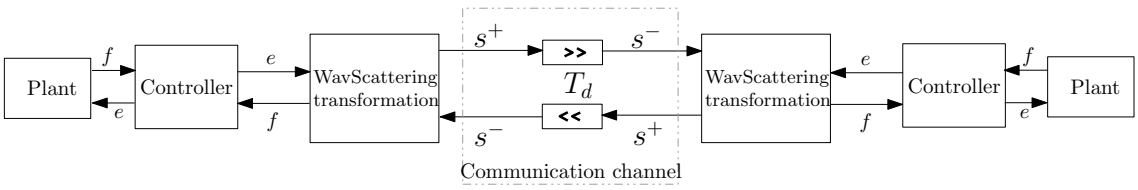


Figure 2.4: Wave scattering transformation in a bilateral telemanipulator.

the power input and power output of a system of which the total power flow into the system consists of. The advantage of this approach is that power flow solely depends on the magnitudes of the waves and is thus unaffected by delays [12]. Hence, methods using wave scattering may make a system robust against time delays. The performance of the total system decreases as the delay increases, but data loss in the the communication is a more critical factor, since its influence on the system performance is more significant.
The association of power input and output of the system with input and output waves is motivated by the physical concept of waves, but can also be applied to any non-linear system [21]. Preserving overall stability using wave scattering does not require any knowledge of the time-delay T_d. When the actual time-delay is minimized to zero, transmitting wave variables is identical to transmitting power variables under the correct causality conditions.
The scattering waves are defined by [25]:

$$s^+ = \frac{1}{\sqrt{2Z}}(-e + Zf) \qquad s^- = \frac{1}{\sqrt{2Z}}(e + Zf) \tag{2.10}$$

Where Z is a positive definite matrix representing the impedance of the scattering transformation. Notice that the signs of e and f in (2.10) depend on the system configuration. The equation set (2.10) describes the configuration depicted in Figure 2.5.
The power balance at the scattering transformation port is represented by:

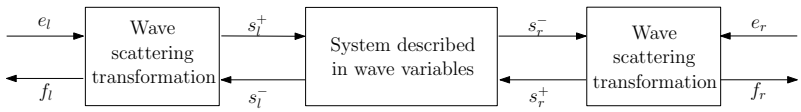


Figure 2.5: Wave scattering transformation

$$P(t) = e^T f(t) = \frac{1}{2}\left\|s_Z^-\right\|^2 - \frac{1}{2}\left\|s_Z^+\right\|^2 \tag{2.11}$$

where $\|.\|$ is properly defined [23] and $\frac{T}{2}\left\|s_Z^-\right\|^2$ and $\frac{T}{2}\left\|s_Z^+\right\|^2$ can be interpret as incoming and outgoing energy packets respectively [26].

The wave transformation is one-to-one and onto. This means that the waves can be inverted to provide the power variables as a function of the wave variables. Moreover, one wave variable together with one power variable is sufficient to uniquely determine each port [21].
Relating wave scattering to passivity, a system is passive if the energy provided by the output wave is limited to the energy received via the input wave (see figure 2.5). Thus the system is passive iff $\exists\beta > 0$ such that [26]:

$$\int_0^t \frac{1}{2}\left\|s_Z^+\right\|^2 d\tau \leq \int_0^t \frac{1}{2}\left\|s_Z^-\right\|^2 d\tau + \beta \tag{2.12}$$

This means that arbitrarily time delays, introduced to the system described by wave variables, will not affect its passivity and hence its stability [21].

Scattering impedance and causality

The positive definite impedance 'Z' can be chosen arbitrarily, keeping in mind that it defines a characteristic impedance associated with the wave variables and is thus of considerable influence on the system response [21]. For instance, wave reflections, which corrupt the useful information and cause oscillatory behaviour, can be avoided by matching the impedance of the attaching subsystem to the impedance 'Z' of the wave transmission.
Matching the wave impedance in order to eliminate wave reflections when the subsystem accepts a flow command to produce an effort (force response) can be done by including appropriate damping

or friction components. However, according to [21], adding terminations at sites where effort is determined by the flow input control causes position drift and thus it is impossible to implement an such a site without including unwanted reflections. Hence, computing f and s^+ as a function of e and s^- is the preferred causality.

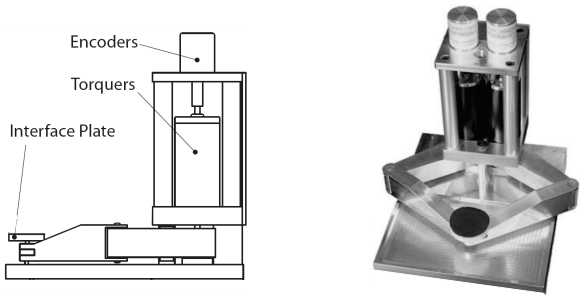
Chapter 3

2 DOF Master-Slave system

The theory discussed in chapter 2 has been implemented in a master-slave system during this project. A short description of the set-up used as the haptic device is given in section 3.1. Section 3.2 expands on the IPC algorithm as it has been implemented. The choices between different alternatives within the algorithm are motivated in this section by means of simulation or measurements. The implementation of the wave scattering formulation is covered in section 3.3.

3.1 Plant

The used set-up in this project is the Pantograph Mk-II designed in the haptic laboratory of the McGill university. This parallel mechanism with 2 DOF can deal with displacements of the order of 10 μm [6]. The pantograph is designed to have very light weight arms to avoid fatigue. This set-up is very backdrivable, which means that the user is able to move the device around without opposition from the actuators.



(a) (b)

Figure 3.1: Pantograph Mk-II

Sensors

The design of this set-up allows the use of position, force and acceleration sensors. The force sensor is designed to be mounted such that force perpendicular to the workspace can be measured. It is chosen not to concentrate on the state changes of the set-up in that direction. Also the acceleration sensor is left out since the controller doesn't make use of acceleration of the interface plate. The only sensors used for this project are two Gurley precision encoders that produce 65536 counts per revolution [22]. The high resolution of these encoders allows highly accurate position

measurements. Appendix D and E give more details about the specifications of the Pantograph, along with its kinematic equations.

Actuators

The actuators on the set-up are two 20W RE25 Maxon motors (graphite brushes) [20]. The motors are connected directly, thus without gearing, to the arms of the Pantograph. Human fingers can exert $30 - 50N$ for brief periods and $4 - 7$ N for sustained periods [2]. $4N$ exerted perpendicular on the shortest arm of the pantograph would result in $0.0785 \cdot 4 = 0.314Nm$ which is too much for the motor connected to that arm since it can only apply a maximum continuous torque of $0.0261Nm$. This means that the human operator can push the mechanism through the most stiff surface simulated by the motor, which makes remote contact with a wall by the slave or a contact with a virtual wall by the master less realistic. Thus limited potential of the motors on the set-up, the geometrical position of the remote/virtual wall and the force applied by the operator determine strongly the stiffness of the surface felt by the user.

Electronics

The motor amplifiers [4] are off-the-shelf hardware available at the CE lab. A PC104 stack [19] is used to be programmed by the controller software. See Appendix I for more details on the used hardware.

Model

The Pantograph has been modelled in the modelling and simulation program 20-sim. For the mass properties of the arms, a mechanical model was made in the 3D CAD software Solidworks. From this model the inertia properties were retrieved that were then used in the 3D model made in 20-sim 3D mechanical editor. The plant model as implemented, can be found in Appendix E, along with the measurements used for the validation of the model.

3.2 IPC algorithm

As explained in section 2.3, an energy based control is the chosen method in this project to guarantee stability of the haptic devices regardless of sample frequency. The most important part of this control algorithm is the determination of the next state. In doing so, there are two main challenges to be faced in the implemented 1D working case;

- determining the sign of the state from an energy level, which is used to choose the next sate from the states corresponding to the same energy level (two states in the 1D case)
- dealing with the passivity losses that can occur between two subsequent sample moments.

A parallel construction of a spring and damper is implemented as the virtual environment. Figure 3.2 visualizes the situation analysed in this section. x and q are expressed in radians. The dimension of the flows is thus radians per second. The other parameters and values are dimensionless (see Appendix H for more detail).

The controller output depends on the state of the system. A constant position of the plant indicates zero energy exchange and thus a steady energy level, which leads to a constant state of the system. When energy exchange occurs, the state will change according to the algorithm explained in the following subsections.

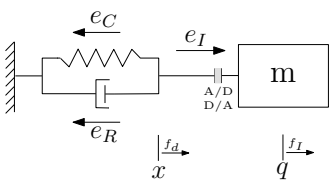


Figure 3.2: The 1D system including a spring-damper controller and a plant

3.2.1 Internal energy

The internal energy that determines the next state of the discrete system, is equal to the integrated sum of all the energy flows. The energy input is the energy exchanged between the continuous part of the system (haptic device) and its discrete counterpart (controller), called ΔH_I. The determination of this energy has been discussed in section 2.3.
A part of the incoming energy is dissipated in order to introduce damping into the controller response, H_R, and to suppress quantization effects, H_Q.

Damping

Damping can be realized using dissipating elements in the controller. In accordance to the mechanical parallel spring and damper construction, the dissipation in the implemented IPC is realized using the derivative of the state of the discrete system; $f_d = f_C = \Delta x/T$. This 'damper' dissipates a portion of the energy coming into the system equal to Tbf_C^2, where b is the damping constant, and contributes to the controller output signal by a factor bf_C. The damper constant is chosen small, which results in an overshoot. This is done to limit dissipation strongly in order to make sure that there is as much energy as possible available in the system to simulate a stiff environment in which the operator feels a high force when trying to move the plant. As Figure 3.4 shows, the displacement of the plant with respect to its original position is very small.

Quantization noise

It was shown in [23] that quantization performed on the measured position can introduce some additional energy to the system and can thus endanger its passivity. In order to eliminate this effect, it is chosen to dissipate the maximum energy production possible by quantization noise $H_Q = e_I \Delta q_{enc}$. This is a very conservative method to deal with quantization errors, but it is at the same time a simple measure to ensure passivity that complies for this project.

3.2.2 Determining the system state

When energy flows into the discrete system, the state of the controller should change to keep the energy balance intact, possibly after compensating the existing energy error by factor α. The value of α is chosen close to its maximum, $\alpha = 1$, to accomplish passivity as fast as possible, which in this case is considered more important than smoother dynamics of the system (see section 2.3).
Of the states that correspond to the determined energy level of the system, the one closest to the current state should be chosen as explained in section 2.3. In the implemented 1D case (see Figure 3.2) there are two states, with the same absolute value but different signs, which have the same energy level. The sign of the next state cannot be retrieved from the energy level due to the quadratic form of the energy function of the linear spring. Therefore the sign is deduced from the position measurement. The sign of Δq shows whether the plant is moving back to its rest position or is moving away from it. The state of the controller and thus the effort exerted to the plant should change proportional to this position shift in order to compensate for the resulting energy flow. This proportionality is achieved by choosing the $x(n+1)$ that leads to the smallest Δx. The smallest Δx means that the shortest route to the next energy level is taken.

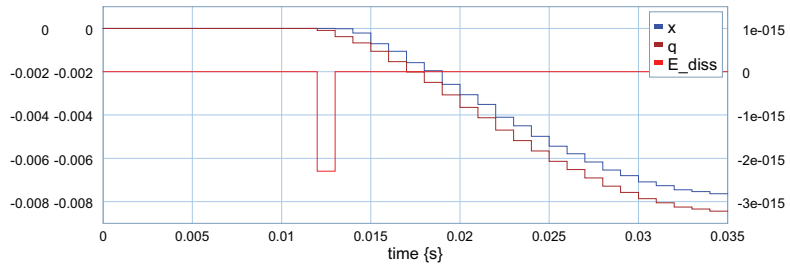


Figure 3.3: Resolving deadlock effect, x and q: [rad]

Deadlock

As explained in section 2.3, the system is not able to change its state anymore once it reaches its minimum energy level. This is due to the zero output of the controller. So by exception, x should be assigned a value such that the system comes out of the deadlock. However, it is important to remember the amount of internal energy that this new state results in ('E_diss' in Figure 3.3), such that it can be dissipated later on, after the deadlock has been resolved.

The disadvantage of this method is drift. The plant has actually moved due to an external force in the continuous domain, when the controller state, x, is assigned a value other than zero. So there has been an unknown amount of energy exchange between the operator/environment and the plant that has caused the plant to move. But the energy exchange at the interaction port of the controller is zero due to the zero controller output (see section 2.3). The controller will not act on this displacement between the moment at which the operator/environment interacts with the plant and the next sample moment, since no energy flow into the controller has occurred. This shift in rest position of the plant can be compensated the next time when a deadlock happens, but it can also increase, depending on the direction of external force on the plant during the next deadlock situation. The latter will increase drift. A more detailed explanation is given here, by looking at the energy functions step by step. As mentioned in section 2.3 $e_I(n)$ is equal to $e_I(n+1)$ at the sample moment. In the following, $e_I(n)$ refers to the effort from the sample moment $t = nT$ until just before $t = (n+1)T$.

Deadlock can lead to a considerable drift. Consider a system as depicted in Figure 3.2, in which the controller is in a deadlock; $x = 0$. Between $t = nT$ and $t = (n+1)T$, a force is initiated on the plant by the operator/environment, e_{ext}. This force leads to a displacement of the plant;

$$|e_{ext}| > 0 \;\Rightarrow\; q((n+1)T) \neq q(nT) \;\Rightarrow\; \Delta q((n+1)T) \neq 0 \tag{3.1}$$

The deadlock is resolved by assigning the controller state x a value based on the position of the plant:

$$\begin{aligned} x(n+1) &= g(q(n+1)) \\ H_C(n+1) &= H(x(n+1)) \;=\; \tfrac{1}{2}kx^2(n+1) \qquad \text{for 1D linear spring} \\ H_{error}(n+1) &= -H_C(n+1) \;=\; -\tfrac{1}{2}kx^2(n+1) \end{aligned} \tag{3.2}$$

The energy error H_{error} should be compensated later. From $t = (n+1)T$ on, the controller is out of deadlock, thus energy exchange at the interaction port of the controller can occur. This H_{error} is only correct if the plant is not in movement before and after the sample moment and its kinetic energy is equal to zero. However, the algorithm only has knowledge about the total Δq during the last sample period. Thus it assumes this energy error to be correct. At this point two cases are possible:

Case 1

$$\begin{aligned} e_{ext}(t) &\neq 0 \qquad \text{for } \; nT < t < (n+1)T \\ \Rightarrow \Delta q(n+1) &\neq 0 \\ \Rightarrow \; e_I(n+1) &\neq 0 \end{aligned} \tag{3.3}$$

From $t = (n+1)T$ on, there is no external force, therefore the movement of the plant is only caused by the controller. This means that the plant moves according to the direction of the controller output, after that the controller output has overcome the momentum of the plant. So energy is flowing out of the discrete system (the controller);

$$\begin{aligned} \Delta H_I(n+2) &= e_I(n+1)\Delta q(n+2) > 0 \\ H_C(n+2) &= H_C(n+1) - \Delta H_I(n+2) \end{aligned} \tag{3.4}$$

The energy error of the previous sample (see 3.2) is also still present as it has not been compensated for yet. Effectively, the system has less energy than it had stored at $t = (n+1)T$, and the energy error is unchanged.
Depending on the value of $\Delta q(n+2)$, $\Delta H_I(n+2)$ can either be greater than $H_C(n+1)$, which would lead to a negative $H_C(n+2)$ and thus an increase of the energy error, or it can be smaller than $H_C(n+1)$, which means that there is less energy stored in the system than at the previous sample moment.

So as the result of taking the controller out of this deadlock, the plant is pushed back for a certain amount, equal to $\Delta q(n+2)$, and the system is left with an energy error that needs to be compensated later. Note that if $\Delta q(n+2) \neq -\Delta q(n+2)$, then the plant has been shifted.
Case 2

$$\begin{aligned} e_{ext}(t) &\neq 0 \qquad \text{for } \; nT < t < (n+m)T \quad \text{where } \; (n+m) > (n+1) \\ \Rightarrow \Delta q(n+1) &\neq 0 \\ \Rightarrow \; e_I(n+1) &\neq 0 \end{aligned} \tag{3.5}$$

At $t = (n+1)T$, there is still an external force on the plant. This means that energy can flow into the discrete system. This is the case when $\Delta H_I(n+2) < 0$. With this energy, the energy error $H_{error}(n+1)$ can be compensated.

$$\begin{aligned} \Delta H_I(n+2) &< 0 \quad \Rightarrow \quad H_C(n+2) > H_C(n+1) \\ H_C(n+2) &= H_C(n+1) - (1-\alpha)\Delta H_I(n+2) = H_{post\ comp.} \\ H_{error}(n+2) &= H_{error}(n+1) - \alpha \Delta H_I(n+2) \end{aligned} \tag{3.6}$$

Thus (a part of) the incoming energy is used for compensating the present energy error. So there is less energy stored in the system than it would have been if there was no energy error. A lower energy level results in a lower controller output:

$$\begin{aligned} H_C(n+2) &= H_C(n+1) - \Delta H_I(n+2) = H_{pre\ comp.} \\ H_{post\ comp.} &< H_{pre\ comp.} \quad \text{and} \quad e_C = \partial H / \partial x \end{aligned} \tag{3.7}$$

$\Delta q(n+1)$ was used to resolve the deadlock the controller was in until $t = (n+1)T$, but the controller never actually acts on this displacement of the plant, since no energy exchange at the interaction port of the controller at $t = (n+1)T$. This displacement between $t = nT$ and $t = (n+1)T$ (when the controller was still in a deadlock), which is a free motion of the plant, causes a drift in the rest position of the plant. The amount of this drift depends on the magnitude of the external force between $t = nT$ and $t = (n+1)T$ and the sample period T.

Another issue is resolving the deadlock. When the controller is in deadlock, the only information available about the continuous world is the position of the plant, q, and the displacement of the plant, Δq. In the ideal case for the system shown in Figure 3.2, where no information is lost and no energy is dissipated, the minimum state of the controller corresponds with the minimum state

of the plant. In this case either q or Δq is appropriate for the choice of the next controller state. However, when the system is driven into deadlock during operation, other considerations apply. q is related to the instantaneous state of the plant. Δq either shows the direction of the net force applied to the plant, or depends on the momentum of the plant. The momentum of the plant is caused by the controller output, which in this case had led to energy production and thus the deadlock in the first place. For this reason, it is the authors believe that if the controller state is set to zero due to an energy error, q is a better indicator of the desired controller response than Δq. A specific example to illustrate this, is a situation in which an external force and an effort by the controller in the opposite direction are exerted to the plant. For convenience, the dissipation is left out. The external force is initially greater than the controller effort, which causes the plant to move in the direction of the external force;

$$\begin{array}{rcl} e_{ext}(t) < 0 & \text{and} & e_I(n) > 0 \\ |e_{ext}(t)| > |e_I(n)| & \Rightarrow & \Delta q(n+1) < 0 \end{array} \tag{3.8}$$

At $t = (n+1)T$, the external force is terminated. At the same time, the controller state is changed based on the exchanged energy at port I;

$$\begin{array}{rcl} e_{ext}(t) & = & 0 \qquad (n+1)T \leq t \\ H_C(n+1) & = & H_C(n) + e_I(n)\Delta q(n+1) \\ H_C(n+1) & > & H_C(n) \qquad \Rightarrow \; e_I(n+1) > e_I(n) \end{array} \tag{3.9}$$

Since there is no external force anymore, at $t = (n+2)T$, Δq changes its direction according to the direction of the controller output (after that momentum is overcome), which is now the only force on the plant. If $\Delta q(n+2)$ is high enough, the system looses its passivity;

$$\begin{array}{rcl} e_{ext}(t) & = & 0 \qquad (n+1)T \leq t \\ H_C(n+2) & = & H_C(n+1) - \Delta H_I(n+1) \end{array} \tag{3.10}$$

$$\begin{array}{rcl} e_I(n+1) > 0 \;\text{ and }\; \Delta q(n+2) > 0 & \Rightarrow & \Delta H_I(n+1) \\ if\; \Delta H_I(n+1) > H_C(n+1) & \Rightarrow & \text{Energy error} \end{array} \tag{3.11}$$

Hence, the system goes into a deadlock; $x(n+2) = 0$. Even though the controller output is zero due to its zero state, $e_I(n+2) = -kx(n+2) = 0$, the plant can keep moving due its momentum.

$$\begin{array}{rcl} e_{ext}(t) & = & 0 \qquad (n+1)T \leq t \\ e_I(n+2) & = & 0 \\ \Delta q(n+3) & > & 0 \end{array} \tag{3.12}$$

If the sample period is long enough, the plant might stop before the next sample moment. At this sample moment, $t = (n+3)T$, the controller state will be assigned a value, in order to come out of the deadlock. If this value is chosen based on the displacement of the plant (using function '$g(\Delta q)$'), the controller will cause a Δq opposite to the one during the last period;

$$\begin{array}{rcl} e_{ext}(t) & = & 0 \qquad (n+1)T \leq t \\ x(n+3) & = & g(\Delta q(n+3)) \\ e_I(n+3) & < & 0 \\ \Delta q(n+4) & < & 0 \end{array} \tag{3.13}$$

This can again cause a deadlock at $t = (n+4)T$ in the same way as at $t = (n+2)T$, thus the energy error increases. This problem (subsequent deadlock) cannot be solved by choosing x based on the position of the plant, q, instead of Δq. However, the plant is pushed back toward its original position $q(0) = 0$, when the position of the plant is used for resolving the deadlock. Depending on the sign of Δq at the sample moment when deadlock is resolved using Δq, the plant is pushed toward, or away from $q(0)$. In other words, the increase of the energy error cannot be prevented, but the position drift of the plant is reduced compared to when Δq is used.

Therefore, $x(n+1)$ should be a fraction of q, small enough not to store a lot of energy. A relatively large x can draw so much energy (especially at low frequencies) that the system becomes unstable, knowing that it should dissipate a high amount of energy, but not getting a chance to do so. With a small x, the energy error to be compensated is small, thus the influence of energy error on the dynamics of the system for the next sample periods will become minimal.

Energy error management

Applying energy leap (see section 2.3), when the negative controller energy level indicates passivity loss, can cause oscillatory behaviour. If the environment/operator does not exert any force on the plant just before and after energy production by the discrete system, the plant will change its position in the opposite direction. Effectively, in this case, by changing the sign of the gradient of the energy function of the controller, and thus the sign of the controller output, the sign of Δq will change as well. This effect can especially occur when the sample frequency is low. Due to the long sample period, there is more time to overcome the inertia and move in the opposite direction. The $\Delta q(n+1)$ will be smaller than $\Delta q(n)$ and have the opposite direction. Since both $\Delta q(n+1)$ and $e_d(n+1)$ have the opposite sign compared with the previous sample time, the sign of energy at the interaction port remains unchanged and thus the energy flow direction stays unaltered. Hence, the system keeps loosing energy and so the energy error increases. Assuming that the ratio between these two subsequent displacements can be given by

$$|\Delta q(n+1)| = \gamma\,|\Delta q(n)| \qquad \text{where } 0 \leq \gamma \leq 1 \tag{3.14}$$

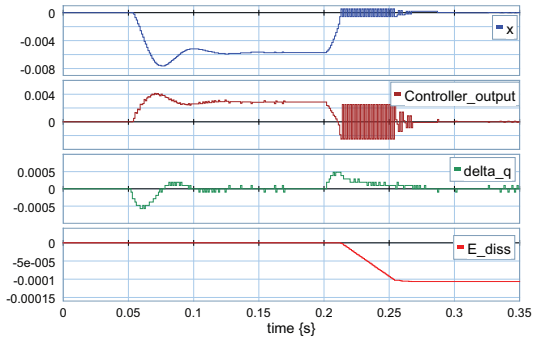
Thus the following will apply:

$$\begin{aligned} \Delta H_I(n) &= e_d^T(n) \cdot \Delta q(n) \\ \Delta H_I(n+1) &= e_d^T(n+1) \cdot \Delta q(n+1) \\ &= -e_d^T(n) \cdot -\gamma \Delta q(n) = \gamma \Delta H_I(n) \end{aligned} \tag{3.15}$$

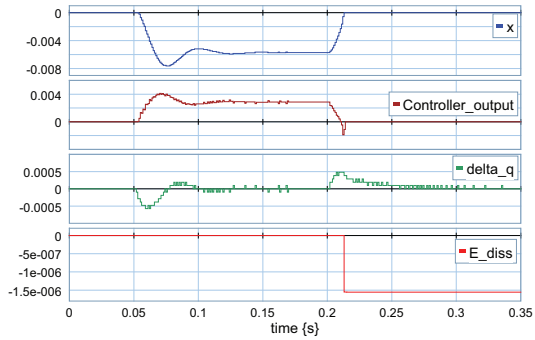
As a consequence, instead of energy absorption, more energy is produced, as shown in (3.15). This leaves the system with a greater energy error, followed by an identical state transition as just performed, and thus more energy production. Due to the energy conserving nature of energy leap and the low friction of the haptic device, the amplitude of this oscillation will decrease very slowly as shown in Figure 3.4.

For high sample frequencies, it is more difficult to change the direction of the plant displacement due to the momentum of the plant in combination with the short sample period. The plant does however get slowed down. Figure 3.4 shows that at high sample frequency, there is indeed an oscillation in the controller output signal, but it does not result in oscillation of the plant. This figure shows that at a low sample frequency, also the position of the plant oscillates. It is clear from this simulation that 'minimum state' method eliminated this oscillation. The relation between Δq and the sign of x is not very clear in this figure, because of the interference of the implemented energy compensation and damper. Appendix B shows the contribution of the energy leap to the oscillating behaviour in absence of the damper and energy error compensation.

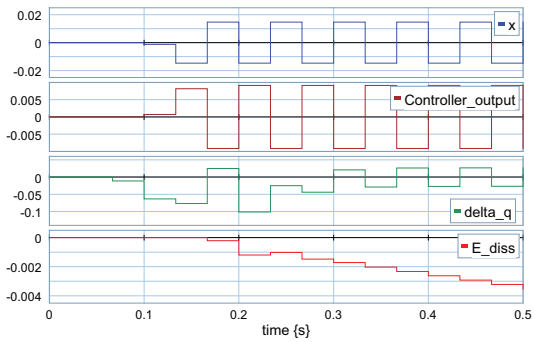
The implemented controller algorithm acts on energy production by setting the state value to its minimum, namely zero. By doing so, the controller uses the available stored energy to compensate for the energy production as much as possible and thus reduces the error to a minimum. This makes the energy error small, which will speed up the passivity recovery. However, the disadvantage of the chosen method is that the system is driven to a deadlock, which always leads to some energy loss as explained earlier in this section. This energy loss interferes with the performance of the system, but does not affect its stability.



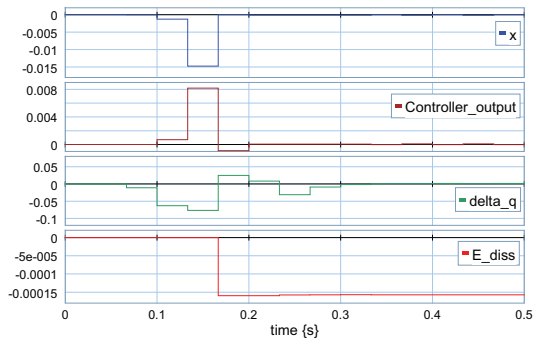
(a) $f_{sample} = 1kHz$, energy leap



(b) $f_{sample} = 1kHz$, minimum energy



(c) $f_{sample} = 30Hz$, energy leap



(d) $f_{sample} = 30Hz$, minimum energy

Figure 3.4: Simulation of two energy error management methods; each figure showing the x $[rad]$, controller output, Δq $[rad]$ and energy error bookkeeping

In [26], the IPC algorithm represented, is summarized by the following steps;

1. Given an initial state $x(n)$, we set $e_C(n) = \frac{\partial H}{\partial x}(x_n)$.
2. Using the value of the system input $f_I(n)$ and the previously calculated $e_C(n)$, we can calculate $e_I(n)$, the output of the interaction port, and $f_C(n)$ using the discrete representation of (3.16) computed at the sampling instant .

$$\begin{pmatrix} e_I \\ f_C \end{pmatrix} = \begin{pmatrix} B & A \\ C & D \end{pmatrix} \begin{pmatrix} f_I \\ e_C \end{pmatrix} \tag{3.16}$$

3. $f_C(n)$ is then used to calculate the next state $x(n+1)$.

In Appendix A, the flowchart of the implemented IPC algorithm during this project is shown. This flowchart follows the same step in the time line, but begins with the third step of the algorithm summery. The implemented algorithm determines $x(n)$ using the energy level of the controller obtained using $f_I(n-1)$ and $e_I(n-1)$. This $x(n)$ determines $e_C(n)$, which is equal to

$$e_C(n) = \frac{\partial H}{\partial x}(x_n) = -kx(n) \tag{3.17}$$

for the 1D implementation here, using a linear spring. Finally, the effort on the interaction port is determined (the block at the bottom of the flowchart in Appendix A), which correspond to the step 2 of the summery. For the implemented system, the following applies:

$$\begin{pmatrix} B & A \\ C & D \end{pmatrix} = \begin{pmatrix} -b & -1 \\ 1 & 0 \end{pmatrix} \tag{3.18}$$

The implemented code for this algorithm can be found in Appendix G.

3.2.3 Virtual environment: wall

As mentioned in chapter 1, rigid environments are in particular very challenging for haptic systems. So the ultimate test for the implemented controller algorithm, is a test involving coming into contact with a stiff surface. For this reason, a test was done with a virtual wall. One of the arms of the Pantograph (decoupled from the other arms) moves freely until it hits the virtual wall at a certain position. After a while the user withdraws the Pantograph from the wall. The expectation was that the set-up stays stable no matter the sample frequency, due to the passivity of the controller. There were two models used for the implementation of the virtual wall; the Kelvin-Voigt model [3] and the Hunt-Crossley model [18]. The first model is very similar to the situation explored in the previous parts of this section. It is first shown how this linear model translates to the practical situation where a virtual wall is envolved. After that, a more realistic behavioural model is used to simulate the presence of a wall; the Hunt-Crossley contact model. It should be noted that these two models are presented for illustrative purposes. The performance obtained by these models are not to be compared as the parameters used are not equal. Furthermore, the force exerted on the set-up was not measured. It is very likely that the force exerted by the operator varies strongly as the contact with the virtual wall is made and the penetration increases.

Kelvin-Voigt contact model

This model is basically the same as the spring-damper model depicted in Figure 3.2 [3]. The behaviour of the wall is modelled by a stiff spring and damper combination. The arm moves freely until it reaches the position of the virtual wall. From that point on, the state of the spring is determined as explained in section 3.2.2. However, instead of the position of the plant, q, the penetration of the plant into the wall, p_w, is used for determining the discrete state of the system. This means that q and Δq are considered to be zero as long as no contact is made with the virtual wall. Figure 3.5 shows the measurement results. Increasing the stiffness of the spring decreases the penetration. However, the virtual wall feels less realistic to the operator because of the vibrations of the set-up arm. This is caused by subsequent overshoots due to the high gain k. These overshoots can be smoothed by applying a higher damping, but then the discrete differentiation used to obtain f_I will contribute to sensing vibrations. When the contact with the virtual wall is ended, an energy

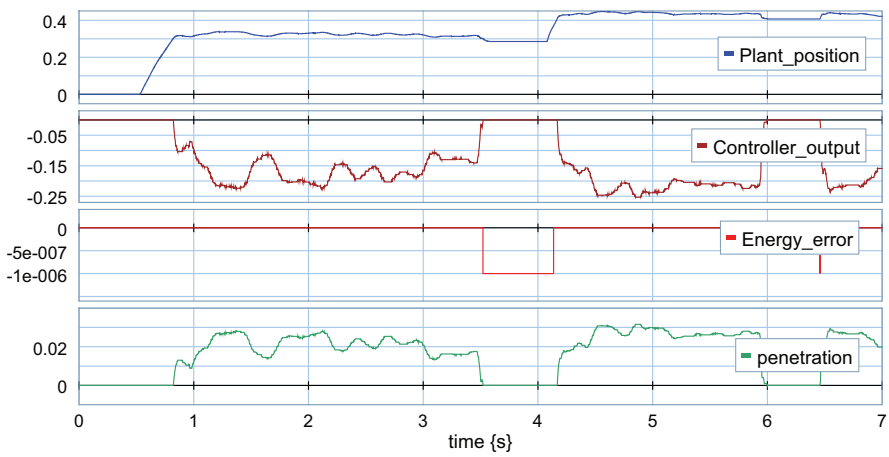


Figure 3.5: Linear contact model, virtual wall at $0.3rad$, $f_s = 1kHz$, $k = 8$, $b = 0.002$, the position and penetration are expressed in radians

error occurs, as explained in chapter 1. This leads to a delay in the controller reaction the next time contact with the wall is made. The controller thus chooses to compensate for the energy error instead of using the incoming energy to control the plant. This contributes to achieving passivity and thus maintaining stability, but at the same time leads to drift.

Appendix C contains more measurement results, which show that the system preserves its stability even with low sample frequencies.

Hunt-Crossley contact model

The non-linear contact model proposed by Hunt and Crossley gives a better behavioural model of a virtual wall. The model also consists of a spring in parallel with a damper, but unlike the previously discussed PD-model, the damper used in the Hunt-Crossley model is non-linear. This non-linear damper, which depends on penetration depth, causes the contact force to build up from zero upon contact, and return to zero on separation. The combination of the spring and non-linear damper results in the following contact force, e_{wall};

$$e_{wall} = -\left(bp_w^{\lambda}\right)\dot{p}_w - kp_w^{\lambda} \tag{3.19}$$

where p_w is the penetration, $\dot{p}_w$ is the penetration velocity and k and b are the spring and damping constant of the wall [18]. λ depends on the surface geometry of the contact. Here, $\lambda = 2/3$ is chosen. This value of λ gives results consistent with Herzian theory on contacting spheres under static conditions [18].

The Hunt-Crossley model gives a good result when it comes to the feeling of hitting a rigid

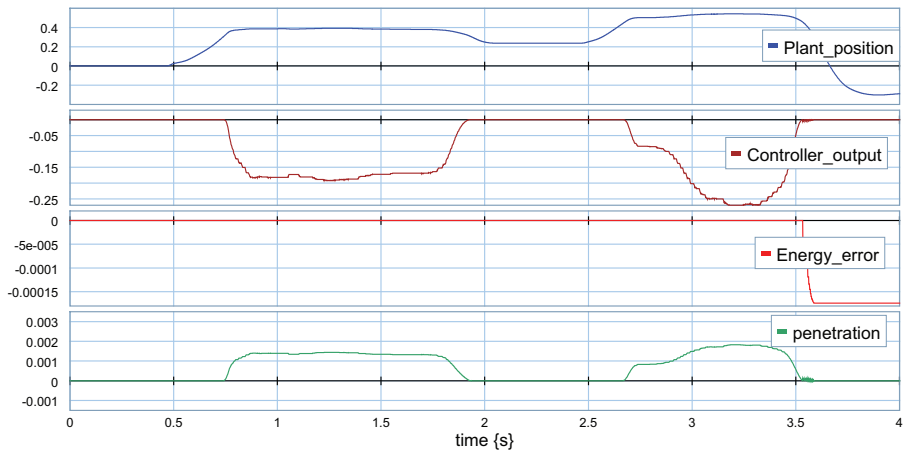


Figure 3.6: Non-linear contact model, virtual wall at 0.4 rad, $f_s = 1kHz$, $k = 3500$, $b = 20$, the position and penetration are expressed in radians

surface. As it can be seen from Figure 3.6, the penetration is quite small. The spring used here is chosen to be much stiffer than in the linear model. Yet no vibrations are felt by the operator when contact with the virtual wall is made. Figure 3.6 shows that the Hunt-Crossley model has a low penetration since it allows a high spring constant. The damping factor is chosen relatively low to preserve energy as explained in section 3.2.1. The drift due to temporary loss of passivity is also visible here. The reader is directed to appendix C for measurements with low sampling frequency.

3.3 Communication over an unreliable channel

In this section the implementation of the scattering algorithm for communication over an unreliable channel is discussed. It should be mentioned that the intrinsic passive control is not included in the model used here. The controller used here is only stable in the appropriate frequency range that it has been tuned for. This controller is only used to include the set-ups in the performed tests. Thus this part of the report focuses only on testing the scattering transformation.

3.3.1 Wave variables transformation

In this project, wave scattering has been implemented for the communication between two identical tabletop robots. Both of these robots are slaves to each other and at the same time function as a master. Each set-up gets a reference signal containing information about the desired flow from the other set-up. This means that both robots simultaneously send their own information to the other one. So there is a symmetry in the total system. For a perfectly symmetric system, the causalities on both sides should be equal. Based on the considerations explained in section 2.4, it is chosen to compute f and s^+ as a function of e and s^-.
The positions of the arms on both robots are measured using high precision encoders. Using this information and the reference signals from the other set-up, the controller determines the effort on the plant. This effort is then used for calculating the output wave that can be decoded to give the reference flow on the other site of the communication channel.

$$s^+ = -\sqrt{\frac{2}{b}}e + s^- \qquad f = \sqrt{\frac{2}{b}}s^- - \frac{1}{b}e \tag{3.20}$$

Because of the symmetry in the system, there is a loop in the information exchange between the end-effectors. Due to the fact that both devices are at the same time masters and slaves, none of the two can initiate an autonomous movement. So the controllers on both sides need information from each other in order to generate an output to the plant and at the same time use the very same output for composing an output wave containing reference for the other plant. This loop is broken by the delay in the communication channel. As long as no data from the master is available, the slave controller receives zero from the channel. For this reason, there should be a delay in the communication line modelled for simulation purposes. There is however no need for this delay in the actual communication, when the communication protocol considers absence of information as generating a zero output. The slave will then assume that there is no energy exchange at the master's side of the system, which is perfectly acceptable since it does not change the total energy present in the system or the energy exchanged between the controllers over the channel.
A lossy channel has been realised by simulating a line in which data packages are lost randomly in a random number (see Appendix F). Also the communication delay varies. This situation emulates unreliable communication protocols over the internet such as UDP. Thus the communication between a master and slave system over the internet can be implemented in a stable fashion using the same method that guarantees the passivity of this simulated communication channel.

3.3.2 Implementation

Scattering transformation has been implemented in combination with a PD controller that operates on the error deduced from the feedback signal and the reference signal received from the channel. Due to quantization in the measured position signal, there is quantization noise present. This noise specially protrudes when calculating the flow, since the division by a relatively small sample time increases the amplitude of the quantization noise. This digital noise can destabilise the system, therefore a Kalman filter is included to filter the noise out of the velocity signal. Refer

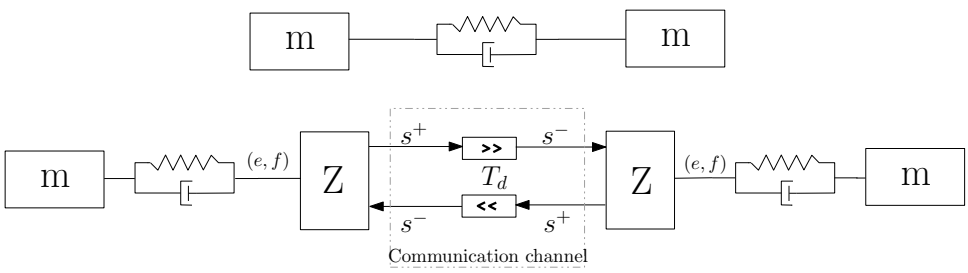


Figure 3.7: Inserting wave scattering in a bilateral manipulator.

to Appendix F for more details on the used model consisting of the wave transformation, controller and Kalman filter.

The realised teleoperator is in effect a spring-damper system split into two parts where the communication using wave scattering is inserted in between the two parts, as shown in Figure 3.7. In case no delay is present, the total system reduces to an impedance controller. The overall system is symmetric; master and slave have admittance causality as they both accept force and convert that to motion (effort in, flow out). The controllers behave as impedances (flow in, effort out) thus the communication is an admittance (to connect two impedances).

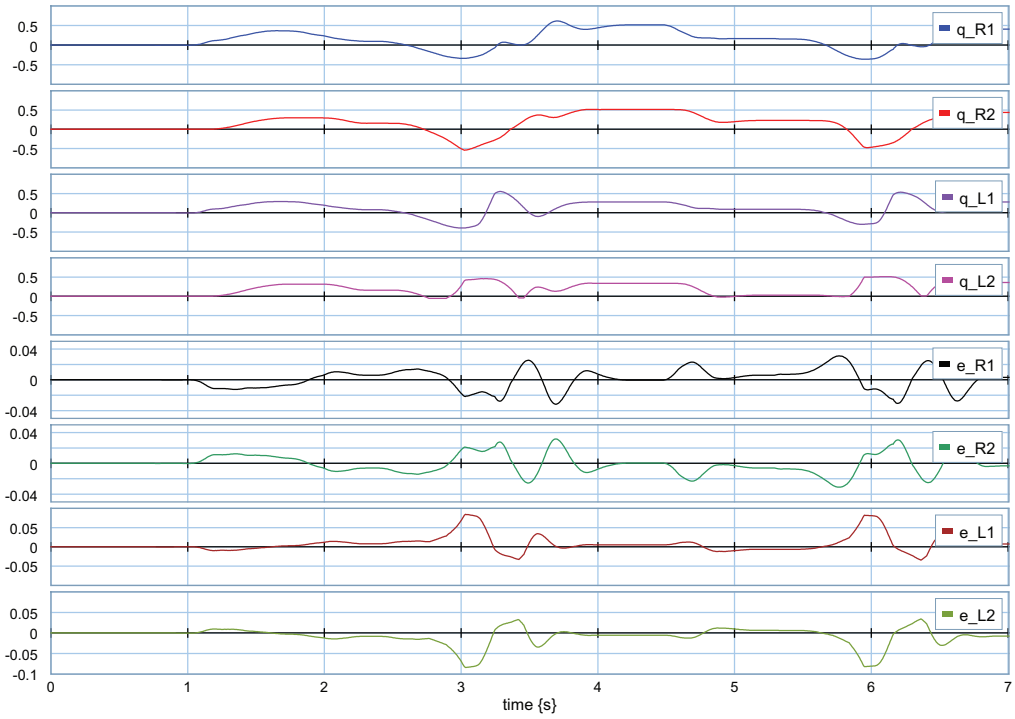


Figure 3.8: Measurements over a channel with variable delay and data loss, using scattering; q: position [rad], L: left arm, R: right arm, e: controller output, 1/2: set-up number.

The measurements depicted in Figure 3.8 were done with random time delays of up to three sample periods, in this case 0.003 seconds, and with data loss in the communication. The results show that the system stays stable. The positions of the set-ups do not match completely due to the data loss, but there is a rough resemblance. The controller output of each set-up is the opposite of its counterpart, which shows that both try to force the robot they control to follow the other one, thus consider their set-up as a slave. When an external force stops, both set-ups move toward each others (changing) position, to finally rest at the same time, when no movement is initiated by the operator anymore and the system thinks that both devices are at the same position. The latter is not true in reality. Due to data loss in the communication channel, the set-ups have different positions.

Chapter 4

Conclusions

4.1 Conclusion

During this project, a set-up designed by McGill university has been built twice, called the Pantograph Mk-II. Using these set-ups a haptic teleoperator can be realized. Off-the-shelf electronics have been used as building blocks for this master-slave system. The total system including the Pantograph has been modelled in 20-sim. This model is validated and used for implementation of the discrete control for the system. Passivity is used in all aspects of the implemented discrete control of the system to guarantee stability.

IPC

An intrinsically passive controller is implemented. Simulation and tests prove the hypotheses of stability of such a controller. As expected, the performance increases as the sample frequency becomes higher.
Energy errors that occur due to the discrete nature of the controller are solved in an energy consistent fashion. The controller compensates for this error as soon as energy flows into the discrete system. The chosen method contributes to achieving passivity as fast as possible. However, it influences the performance of the system dynamically.
Deadlock in the controller state has been resolved by using the state change of the plant, which indicates energy exchange between the plant and the the environment. Due to zero output of the controller, a part of this energy exchange is neglected. This is done because due to the zero output of the controller, there is no energy exchange at the interaction port of the controller. Compensating the total energy that is introduced to the system while coming out of a deadlock, guarantees that no energy is produced by the controller. However, it also means that the controller might have less energy available than what is used by the operator and or the environment on the plant. As a consequence, the controller is unable to recover the plants original state. This can lead to drift in time, especially at low sample frequencies.

Wave scattering

The scattering formalism has been implemented in combination with a PD controller acting on the position error. The results are very promising. The tests show that stability is maintained as (variable) communication delays arise. The system's stability is also insensitive to data loss. The more data packages are lost in the communication channel, the lower the haptic performance of the system. However, the system preserves its stability.

The overall conclusion is that the ingredients for a stable haptic bilateral telemanipulator have been implemented and tested separately. The results match the theory. Where it comes to

performance, a lot can be gained. Due to lack of time, the total system consisting of the IPC and wave scattering transformation has not been implemented.

4.2 Discussion

There is a drift problem as discussed in chapter 3. Drift can occur when data is lost in the communication between the haptic devices. Another reason can be energy loss due to deadlock. This problem can be solved by a supervisory system that injects energy into the system when necessary. This is of course a very delicate task. When energy is injected into the system by the supervisor, the total system looses its passivity. The low-level spring-damper controller keeps its passivity, but the overall system will be active since energy is 'produced' by the supervisor. Thus it is very important that energy injection is done in such a way that the system remains stable despite loosing its passivity. The supervisor can use the state of the haptic device(s), which can indicate drift, and use that in combination with other factors like sample frequency, to determine the amount of energy injection that is 'allowed'. This will increase the performance of the system.
Section 3.2.1 shows that high damping dissipates more energy, which is disadvantageous for the total performance of the system, for instance by causing drift due to insufficient controller internal energy. The supervisor can thus also keep track of energy dissipation by the damper and use this information when injecting energy into the system, to improve the system performance. This allows the implementation of higher damping for better system dynamics.

In case of passivity loss as discussed in 3.2.2, the energy error is partially limited by driving the system into a dynamic deadlock. This eventually leads to energy loss while deadlock is being resolved. It was shown that the dynamic performance of the system is better in this case compared to energy leap, but it is still far from ideal. A more intelligent solution should be found that avoids the deadlock, yet still manages the energy error limited. An alternative would be to decrease the controller state, not to its minimum, but to a sufficiently low level. Also the sign of the next state can be altered, as it would in case of applying energy leap. The correct mix of energy leap and the currently implemented method is subject to more investigation.

4.3 Recommendations

The first recommendation concerns the unfinished parts of this project. The focus of this project was on implementing a passive system. The proposed concept has been validated by tests for both wave scattering transformation and the IPC. However, these concepts have not been tested in the actual setting where they will be used. In the actual setting, there will be an internet connection between the devices instead of a simulated unreliable communication channel. For this, a communication protocol that can be used for communication between two PC104's, should be implemented. Also the complete set-up (2 DOF) should be controlled by the implemented IPC in combination with the scattering transformation.
The 20-sim model of the plant can be improved by modelling frictions in the joints. This is especially recommended when the complete set-up is used. As it can be seen in Appendix E, the controlled model of the plant shows strong similarities between the simulation and measurements. However, the unmodelled frictions cause an error in open loop comparison.
In addition to testing the system in its actual setting, scattering formalism can be implemented for the ZOH in the system in order to sustain stability despite possible computational delays. This contributes to guaranteeing stability no matter the conditions the system is in.

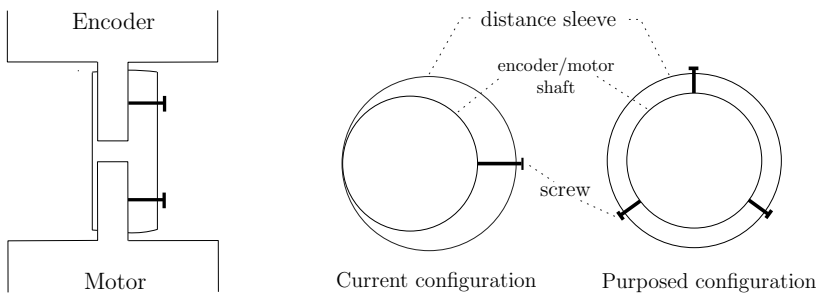
No homing action has been implemented in the control algorithm yet. This means that a difference between the position of the master and slave can exist, which remains unchanged during the operation of the system. It is also recommended to include safety measures in the algorithm based on the performed homing, such that the set-up never reaches its mechanical limits. This

protects the set-up mechanically.

In the current IPC algorithm, the quantization noise is resolved in a very conservative fashion as explained in section 3.2.1. Effort should be put in getting a better picture of the actual energy production due to quantization noise so that a more realistic compensation can be implemented to reduce energy dissipation. In [23] a few alternatives are suggested that can be applied.

There is also room for improvement concerning the used electronics. A current limiter on the H-bridge will protect the motor. In the current setting, the power supply is limited to the maximum continuous current allowed for the used motor [20]. However, it is recommended to implement this limit in the hardware used such that the motor protection does not depend on the used power supply. The H-bridge used at this moment has a current limiter of 10 A. This limit can be adjusted to the maximum continuous current of 1.17 A for the Maxon RE25. However, to get a better haptic result, limiting the current through the motor should be done in a smart way. In order to give the operator a realistic feedback in case of remote collisions with highly stiff (virtual) objects, it is better to allow a peak force at the contact moment. This can be done by allowing the absolute maximum current the RE25 can have, for brief periods. This can partially compensate the fact that the motors are not as strong as human fingers (see 3.1).
Another recommendation concerning the electronics is custom designing electronics that can replace the used over-dimensioned H-bridge and switching board. This will result in space efficiency, which is desired in a demonstration set-up.
It is also recommended to use real current amplifiers (instead of H-bridges) to obtain a linear relation between the output of the controller and the applied torque on the set-up.

In order to improve the set-up mechanically, the assembly of the encoders on the pantograph motors should change. Currently, the encoder is mounted on the motors by means of a distance sleeve. The distance sleeve is put against both the encoder shaft and the motor shaft using two screws (see Figure 4.1). This assembly is sensitive to alignment, which can form a problem when the set-up is taken apart and remounted. A better bushing or a more symmetrical construction (for instance using six screws) can resolve this problem.



(a) Current assembly, side view

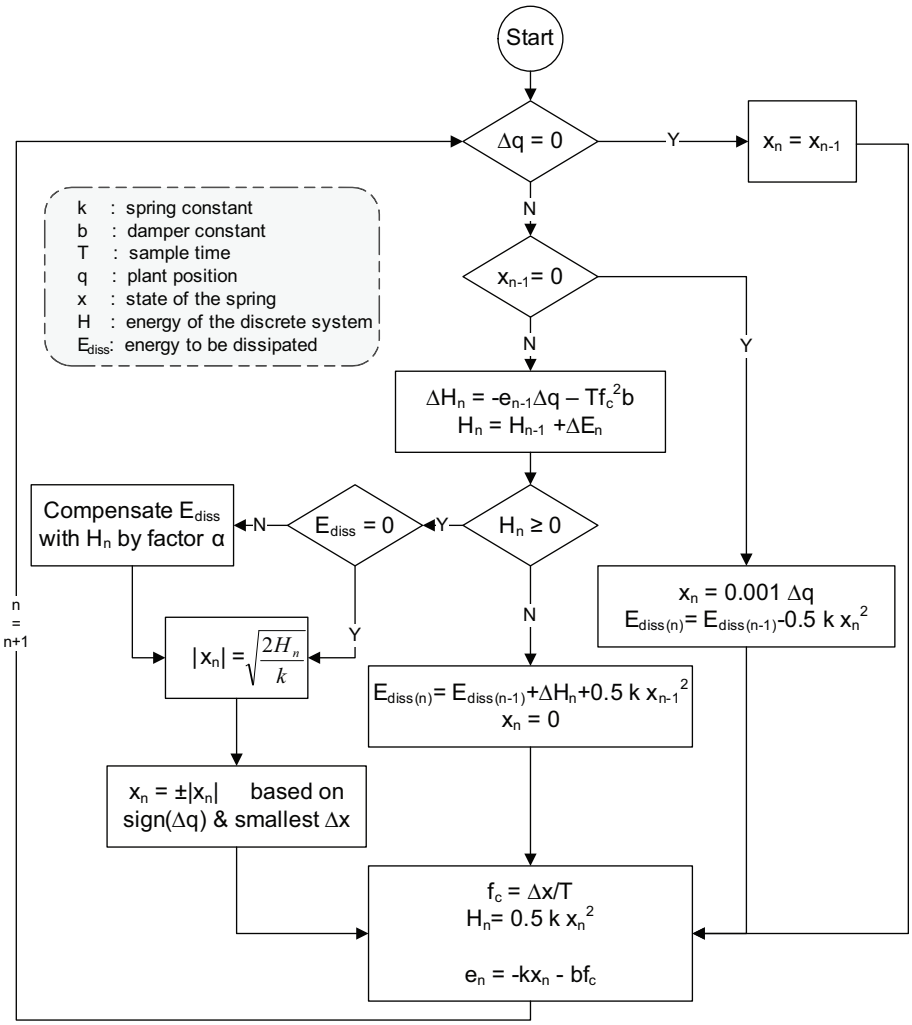
(b) Top view

Figure 4.1: The assembly of the encoders on the motors

Appendices

Appendix A

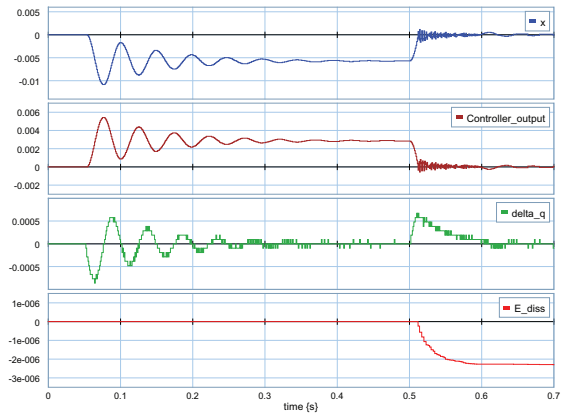
IPC flow chart



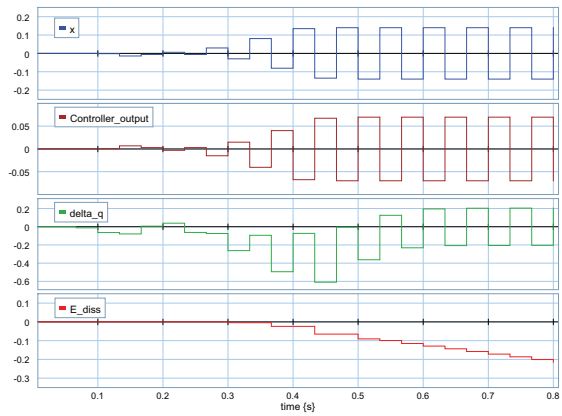
Appendix B

Energy leap

Figure B.1 shows the contribution of the energy leap to the oscillating behaviour, which is amplified by the damper and energy error compensation in a latter stage as depicted in Figure 3.4.



(a) $f_{sample} = 1kHz$



(b) $f_{sample} = 30Hz$

Figure B.1: Simulation showing the oscillating effect of energy leap; each figure showing the controller state x, controller output, Δq and energy error bookkeeping

Appendix C

Virtual wall

Figure C.1 and C.2 show that the haptic device remains stable when in contact with a rigid environment (virtual wall), even with low sample frequency (30 and 1 $[Hz]$). For both measurement, the Hunt-Crossly model was used [18].

Figure C.1 shows that when the robot arm is withdrawn from the virtual wall, energy is produced

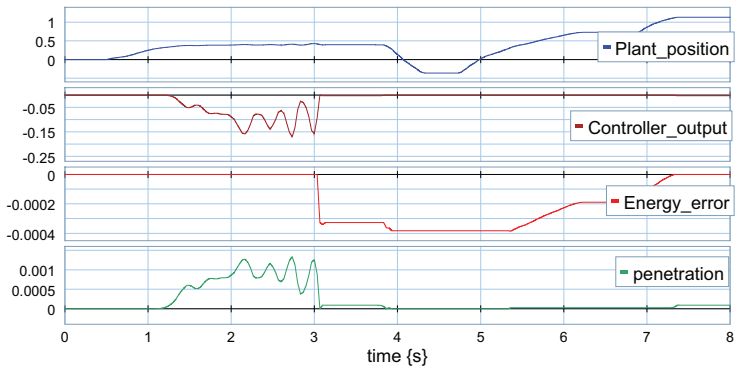


Figure C.1: $f_s = 30\ Hz$

(thus the energy error increases). Next time contact is made with the wall, the controller uses the incoming energy to compensate for this error. This effectively appears to the operator as if a position shift of the virtual wall has occurred.

When sample frequency is only 1 $[Hz]$, there is less room for compensating the energy error since

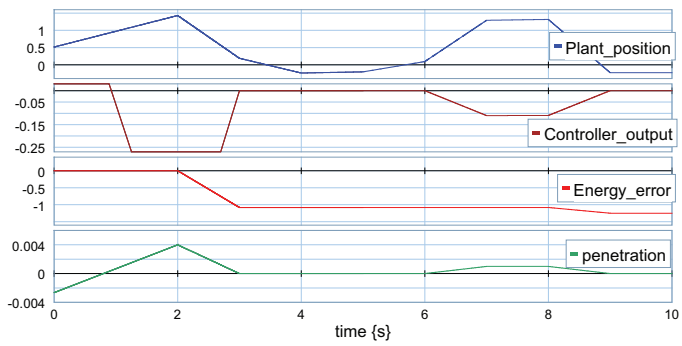


Figure C.2: $f_s = 1\ Hz$

both times the contact is very brief. As it can be seen in Figure C.2, the energy error caused when

the first contact with the wall is ended, is increased by the second contact. This is due to the method applied to resolve the deadlock the controller is in. Before the controller has a chance to compensate the accumulated error, one sample later, the contact is already broken. This behaviour of the operator leads to energy production by the controller.

Appendix D

Kinematics of the Pantograph

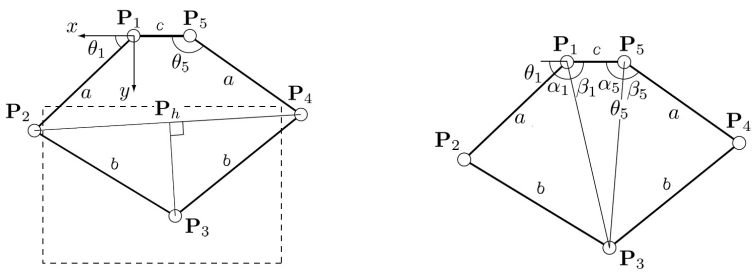
D.1 Direct kinematics

$\mathbf{P}_3$ is at the intersection of two circles that are centered at $\mathbf{P}_2$ and $\mathbf{P}_4$. The z axis passes through $\mathbf{P}_1$ [6].

$$\begin{aligned}
\mathbf{P}_2(x_2, y_2) &= [a\,cos(\theta_1),\ a\,sin(\theta_1)]^T \\
\mathbf{P}_4(x_4, y_4) &= [a\,cos(\theta_5) - c,\ a\,sin(\theta_5)]^T \\
\\
\|\mathbf{P}_2 - \mathbf{P}_h\| &= \tfrac{1}{2}\,\|\mathbf{P}_4 - \mathbf{P}_2\| \\
\|\mathbf{P}_3 - \mathbf{P}_h\| &= \sqrt{b^2 - \|\mathbf{P}_2 - \mathbf{P}_h\|^2} \\
\\
\mathbf{P}_h &= \mathbf{P}_2 + \frac{\|\mathbf{P}_2 - \mathbf{P}_h\|}{\|\mathbf{P}_2 - \mathbf{P}_4\|}\,\|\mathbf{P}_4 - \mathbf{P}_2\|
\end{aligned} \tag{D.1}$$

The mechanical construction of the Pantograph is limited such that it operates in the configuration with the largest y. Keeping in mind that $x_4 < x_2$ in the workspace of the Pantograph, $\mathbf{P}_3$ becomes:

$$\mathbf{P}_3(x_3, y_3) = \begin{cases} x_3 = x_h + \dfrac{\|\mathbf{P}_3 - \mathbf{P}_h\|}{\|\mathbf{P}_2 - \mathbf{P}_4\|}(y_4 - y_2) \\ y_3 = y_h - \dfrac{\|\mathbf{P}_3 - \mathbf{P}_h\|}{\|\mathbf{P}_2 - \mathbf{P}_4\|}(x_4 - x_2) \end{cases} \tag{D.2}$$



(a) Direct kinematics (b) Inverse kinematics

Figure D.1: Schematic model of the Pantograph Mk II

$$a = 6.3[cm], \qquad b = 7.5[cm], \qquad a = 2.5[cm]$$

D.2 Inverse kinematics

$$\theta_1 = \pi - \alpha_1 - \beta_1, \qquad \theta_5 = \pi - \alpha_5 - \beta_5$$

$$\begin{aligned} \alpha_1 &= arccos\left(\frac{a^2-b^2+\|\mathbf{P}_1,\mathbf{P}_3\|}{2a\sqrt{\|\mathbf{P}_1,\mathbf{P}_3\|}}\right) & \beta_5 &= arccos\left(\frac{a^2-b^2+\|\mathbf{P}_5,\mathbf{P}_3\|}{2a\sqrt{\|\mathbf{P}_5,\mathbf{P}_3\|}}\right) \\ \beta_1 &= atan2(y_3, -x_3) & \alpha 5 &= atan2(y_3, x_3 + c) \end{aligned} \tag{D.3}$$

The solution is valid only if $\alpha_1 > 0$ and $\beta_5 > 0$. Otherwise the device would be out of its workspace [6].

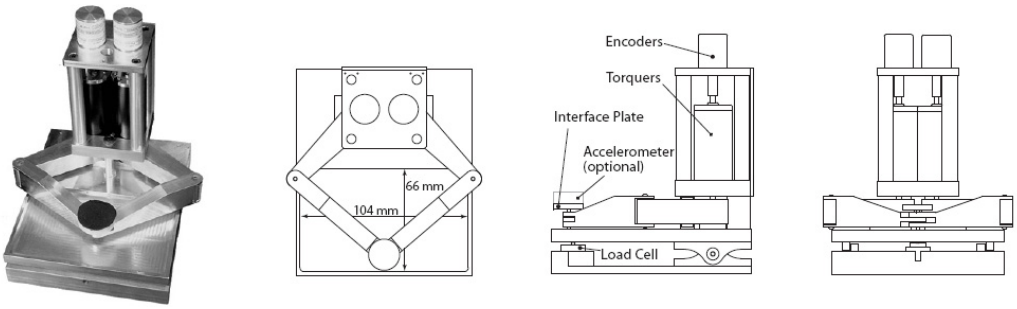


Figure D.2: Pantograph Mk II

Appendix E

20-Sim model of the plant

E.1 Mechanics

The Pantograph has two identical upper arms and two lower arms that are very similar to each other. For convenience, one model is used for the lower arms. Since the shape and weight of the lower arms are very alike (see Figure E.2), this approximation is valid.

Upper arm

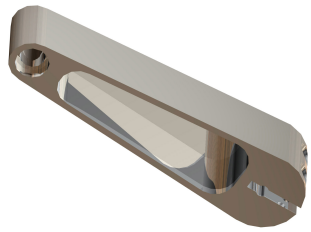


Figure E.1: The upper arm

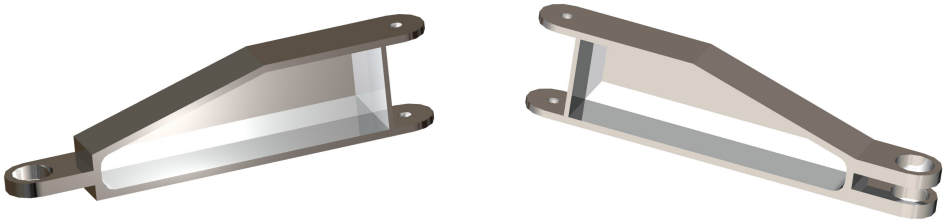
Mass = 27.60 gr
Principal axes of inertia and principal moments of inertia $[gr \cdot mm^2]$ taken at the centre of mass:

$$\begin{array}{rclrcl} I_x & = & (1.00,\ -0.01,\ 0.00) & P_x & = & 1523.72 \\ I_y & = & (0.00,\ 0.00,\ -1.00) & P_y & = & 17199.42 \\ I_z & = & (0.01,\ 1.00,\ 0.00) & P_z & = & 17413.12 \end{array} \tag{E.1}$$

Lower arm

Both arms together weight 25.8 gram. So each arm is modelled with a mass equal to 12.95gr.
Principal axes of inertia and principal moments of inertia $[gr \cdot mm^2]$ taken at the centre of mass:

$$\begin{array}{rclrcl} I_x & = & (1.00,\ 0.09,\ 0.01) & P_x & = & 717.30 \\ I_y & = & (-0.09,\ 1.00,\ 0.01) & P_y & = & 7492.70 \\ I_z & = & (-0.01,\ -0.01,\ 1.00) & P_z & = & 7893.84 \end{array} \tag{E.2}$$



(a) (b)

Figure E.2: The lower arms

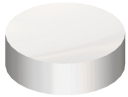


Figure E.3: Interface plate

Interface plate

Mass = 4.2 gr
Principal axes of inertia and principal moments of inertia $[gr \cdot mm^2]$ taken at the centre of mass:

$$\begin{array}{rcl} I_x & = & (0.00,\ 0.00,\ 1.00) \\ I_y & = & (1.00,\ 0.00,\ 0.00) \\ I_z & = & (0.00,\ 1.00,\ 0.00) \end{array} \qquad \begin{array}{rcl} P_x & = & 200.09 \\ P_y & = & 200.09 \\ P_z & = & 355.32 \end{array} \tag{E.3}$$

The Pantograph Mk-II

The model of the Pantograph is made in the 3D mechanical editor of 20-sim. The constraint joint on which the interface plate is mounted, is modelled as a parallel construction of a stiff spring-damper connection. A 20-sim model is generated automatically from this model, which is connected to the Maxon motor model in 20-sim.

Maxon Motor

The model represents the ideal situation in which frictions, thermal effects and motor efficiency are neglected.

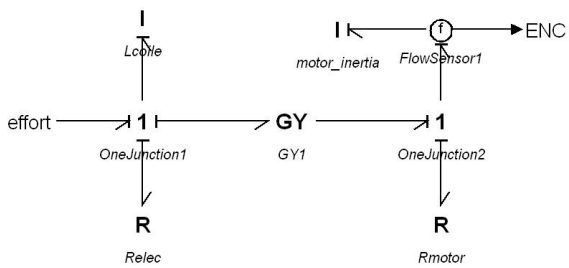


Figure E.4: Bond Graph model of the maxon RE25

Parameter	Value	Unit
R_{elec}	2.32	Ω
I_{Lcoile}	0.24	mH
GY	23.4	mNm/A
R_{motor}	$2.14e-7$	Ω
$I_{motor\_inertia}$	10.3	gcm^2

Table E.1: Motor parameters

Gurley encoder

The encoder model receives the flow value of the motor. This flow is first integrated to obtain the position in radians. Then, a gain is used to model the output of the Gurley encoders, which produces 65536 counts per revolution.The code of this gain is shown here;

```
// radians / (2.Pi) = number of revolutions
// number of revolutions . 65536 = total number of pulses
// k = 65536 / (2.Pi)
equations
     output = input*65536/ (2*pi);
```

The output is then truncated by 1 since the real encoder has multiples of one as output.

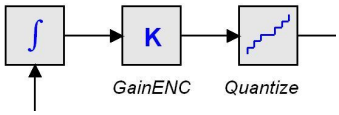


Figure E.5: Encoder model

E.2 Electronics

Encoder Interfaces

The encoder interface converts the encoder signal to radians.

```
//total number of pulses / 65536 = number of revolutions
//nr. revolutions . (2 . Pi) = radians
// k = 2.Pi / 65536
equations
     ENCIoutput = input * 2*pi / 65536;
```

H-bridge

The H-bridge has been modelled as gain equal to 24. The H-bridge input is a normalized value produced by the controller between -1 and 1, where the sign represents the direction of the PWM and the absolute value indicates the duty cycle of the PWM signal; $input = 1 \Rightarrow 100\%\ PWM$ with an amplitude of $24V$. The relation between the voltage on the motor and the PWM signal on the H-bridge is assumed to be linear.

E.3 Validation

The model of the total system is linearised as mentioned. Also some effects have been neglected, the most important of which are the frictions. This causes considerable deviation between the

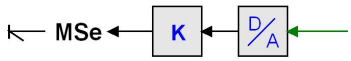


Figure E.6: The H-bridge model

simulations and measurements in open loop. In closed loop situation, the error reduces to an acceptable level.

E.3.1 Motor model

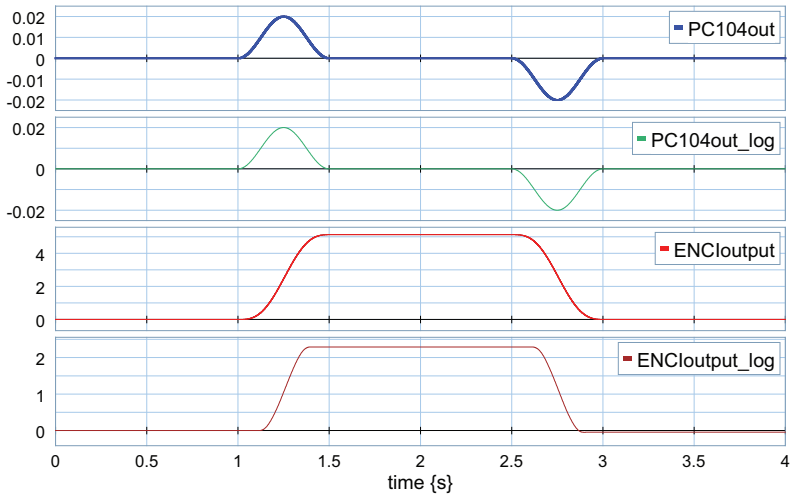


Figure E.7: Validation Maxon motor, no load; PC104 output:reference flow $[rads^{-1}]$, encoder interface output: position$[rad]$; log: logged signal (measurement)

The motor model is based on the parameters given by Maxon Motor [20]. The errors in the simulation results and measurements are mainly caused by the linearised model of the H-bridge and frictions that are not modelled.

E.3.2 Set-Up

One arm

The results of the test with one arm show that the deviation between the simulation and the measurement. This means that the model of the arm is more realistic than that of the motor and the PWM generation. The motor loaded with an arm functions more similar to the simulation. Again, there are some differences between the measurements and the simulations, due to approximations and simplifications made.

The Pantograph MK-II

As mentioned before, the neglected frictions in the complete model and linear approach in the H-bridge model, cause differences between the measurement results and the simulations. The frictions are higher when the total set-up is used instead of only one arm, since four joints are added to the set-up. This means that the motion profile used for validation should have a high amplitude so that the actual set-up makes a movement that is not to small. But on the other hand, the larger the

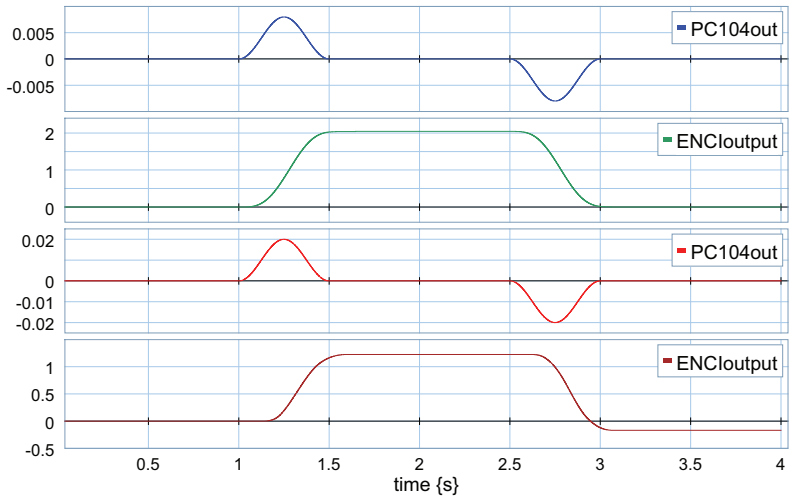


Figure E.8: Validation, 1 arm; PC104 output:reference flow $[rads^{-1}]$, encoder interface output: position$[rad]$; upper two: simulation results, lower two: measurement

movement made in simulation, the more difficult for 20-sim to maintain the modelled constraint joint together. This means that only low amplitudes should be used for open loop validation. The disadvantage of small amplitudes is that the set-up arms only move a little. The static friction is then, which increases the difference between the simulations and the measurements, since it is not modelled. For this reason, the set-up is validated in close loop; that is with a controller that compensates for the linearity assumptions and neglecting frictions. Figure E.9 shows the results. The position of the arms connected to the motors are measured. This position is actually the angle of the arm in radians. It is clear that the close-loop model strongly resembles the real set-up.

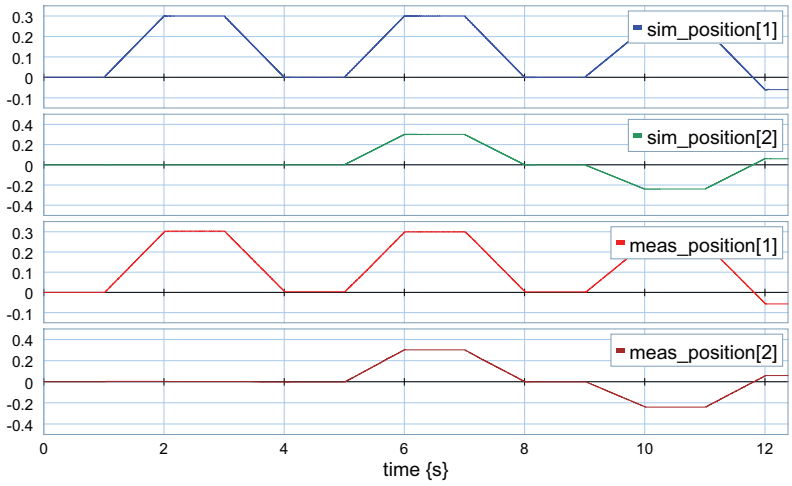


Figure E.9: Validation of the set-up; sim: simulated value, meas: measured value

Appendix F

20-Sim model of the wave scattering

Figure F.1 shows the 20-sim model in which the wave scattering transformation is implemented. Implementing the wave scattering transformation and the controller separately causes an algebraic loop that cannot be solved by 20-sim. By combining the equations of these two blocks, the loop in the model was solved manually.

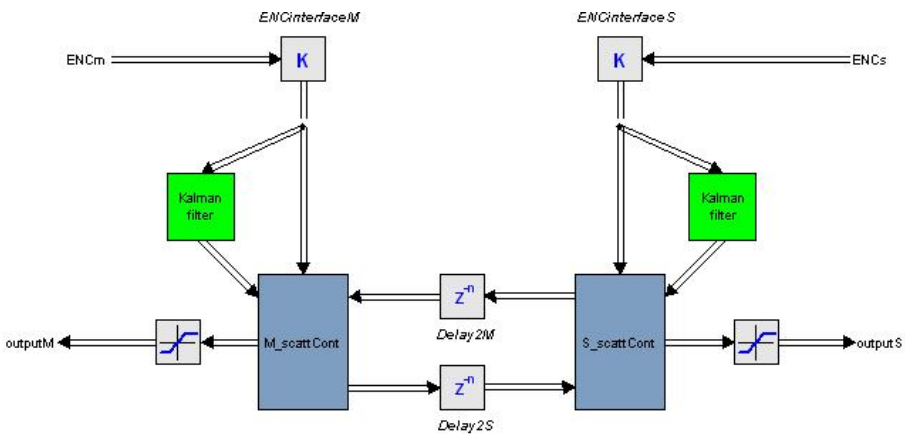


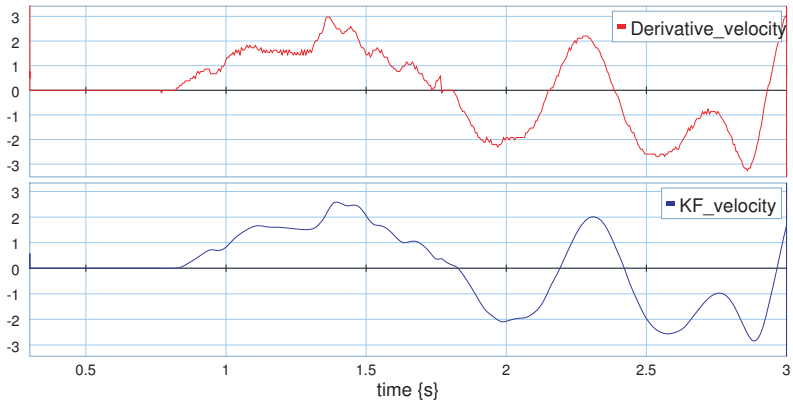
Figure F.1: The 20-sim model of the implemented wave scattering transformation

F.1 Kalman filter

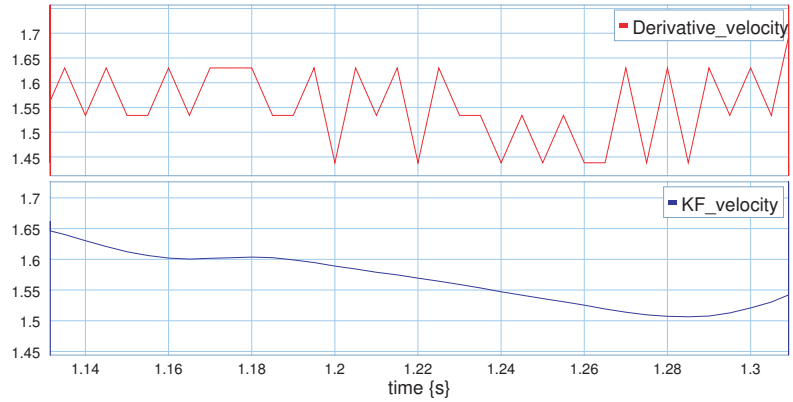
Figure F.2 shows the measurements that validate the effect of the Kalman filter. The quantization noise is clearly filtered out. Please note that here, x is a local variable and doesn't correspond to the state of the IPC.

z : Measurement
x : State vector
n : Measurement noise
w : Process noise

$$\begin{aligned} \mathbf{x(i+1)} &= F \cdot \mathbf{x(i)} + \mathbf{w} \\ z(i) &= H \cdot \mathbf{x(i)} + \mathbf{n} \end{aligned} \tag{F.1}$$



(a) Normal view



(b) Zoomed in

Figure F.2: Measurements showing: Derivative_velocity: velocity obtained by differentiating the measured position; KF_velocity: the predicted velocity by the Kalman filter

$$z = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ f_1 \\ f_2 \end{bmatrix}$$

$$\mathbf{x}_{i+1} = \begin{bmatrix} x_{i+1} \\ \dot{x_{i+1}} \end{bmatrix} = \begin{bmatrix} x_i + \Delta T \cdot \dot{x_i} \\ \dot{x_i} \end{bmatrix} + \begin{bmatrix} 0 \\ w_{\dot{x_{i+1}}} \end{bmatrix} \tag{F.2}$$

$$F = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{F.3}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{F.4}$$

Measurement noise

$\mathbf{n} = \sigma_n^2$
C_{meas}: Measurement noise covariance matrix, $[2 \times 2]$

σ_n: $error_{encoder} = error_{instrument} + error_{quadrature} + error_{quantization} + error_{interpolation}$

$error_{instrument} = 4\ [arcminutes] = 4/60\ [deg]$
$error_{quadrature} = 24\ [deg]$
$error_{quantization} = \frac{2\pi}{2 \times 65536}[rad]$
$error_{interpolation} = 0$ since no interpolation is applied

$\sigma_n = (\frac{\pi}{65536}) + [(\frac{4}{60} + 24) \times 2 \times \frac{\pi}{360}] = 0.176476037$

$$C_{measurement} = \begin{bmatrix} 0.176476037 & 0 \\ 0 & 0.176476037 \end{bmatrix} \tag{F.5}$$

Process noise

$\mathbf{w} = \sigma_w^2$
$C_{process}$= Process noise covariance matrix, $[4 \times 4]$
The implemented Kalman filter assumes that the velocity doesn't change between two subsequent sample moments (other than the white noise that is added to it). This is of course not true. For this reason the uncertainty of the estimated velocity is chosen high. This uncertainty also affects the predicted position. Due to the fact that velocity is integrated to obtain the next position, the error in the position prediction would be relatively small. The value of the position varies in time, but not as strong as the velocity. This makes the uncertainty in the position value relatively smaller when compared to the velocity. Notice that only the flow estimated by the Kalman filter is used in the controller. For the position, the measured value is used. The implemented Kalman filter is tuned using measurements, by visually smoothing the signal and the same time trying not to loose information about rapid changes in the velocity. The result is the following covariance matrix for the process noise:

$$C_{process} = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \tag{F.6}$$

20-sim code

```
variables
      real global  Fs[4,4];
      real Kg[4,2], S[2,2], innov[2];
      real xpre[4], cpre[4,4], xest[4], cest[4,4];

parameters
      real global cmeas[2,2]    = [0.176476037, 0.0;0.0, 0.176476037];  //sensor
          noise
      real global cprocess[4,4] = [0.01, 0.0, 0.0, 0.0;0.0, 0.01, 0.0, 0.0;
                0.0, 0.0, 10.0, 0.0;0.0, 0.0, 0.0, 10.0];     //process noise
      real global H[2,4] = [1.0, 0.0, 0.0, 0.0;0.0, 1.0, 0.0, 0.0];

initialequations
      xpre = [0; 0; 0; 0];
      //predictation covariance matrix, high since no homing is done now
      cpre = [1,0,0,0; 0,1,0,0; 0,0,1,0; 0,0,0,1];
      Fs = [1,0,sampletime,0; 0,1,0,sampletime; 0,0,1,0; 0,0,0,1];

equations
```

```
//update
S  = H * cpre * transpose(H) + cmeas;    //innovation covariance
Kg = cpre * transpose(H) * inverse(S);   //Kalman gain
innov = pos - H * xpre;   //innovation matrix

xest  = xpre + Kg * innov;
cest  = cpre - Kg * S * transpose(Kg);

// prediction
xpre = Fs * xest;
cpre = Fs * cest * transpose(Fs) + cprocess;

f = [xpre[3]; xpre[4]];
```

F.2 Scattering and PD-controller

```
variables
     real posr[2], posError[2], fr[2], fError[2];
parameters
     real global k= 0.2;        //spring constant
     real global b= 0.01;            //damper constant
     real global ffg= 0.001;   //feed forward gain

equations
//................Scattering
Sp = -(sqrt(2/b) * output)+Sm;
fr = sqrt(2/b)* Sm-(output/b);

//................Controller
next(posr) = posr + fr*sampletime;
fError = fr-f;
posError = posr - pos ;

output = (ffg*sqrt(2/b)*Sm + k*posError +b*(sqrt(2/b)*Sm-f)) / (2 + ffg/b);
```

F.3 Variable delay

The data on the communication channel is put into an array. The difference between the array indexes of two subsequent data packages is randomly chosen between 1 and a predefined number, *maxd* (delay). The array is read at the output. Depending on whether there is an input in each array cell, the output would be zero or an actual received data. When the array is full, data is put in from the beginning of the array, resulting in erasing not only the already read data, but also the data that have not been put on the output (data loss). The 20-sim code of this delay is listed below;

```
parameters
     real index = 1000.0;
     real maxd = 1;

variables
     real u, array[index], i, j, n, flag;

initialequations
     i = 1;
     j = 1;
     n = 1;
     flag = 0;
     array =0;

equations
```

```
output = array[j];
array[j] = 0;        //data already used => wipe it out of the array
array[i] = input;

//increase the counters
  u = trunc(ran(maxd)+2);      //determines the maximum delay
  i = u + previous(i);    //random increase to simulate variable delay
  j = previous(j) + 1;

//reset the counters when the end of the array is reached
if i>index then
     i = 1;
end;
if j>index then
     j = 1;
end;

//data loss: occurs when the array is full, but it has not been read completely
if i<j then
     flag = 1;
     for n = i to (j-1) by 1 do
          array[n] = 0;
     end;
else
     if flag == 1 then
          flag =0;
          for n = i to index by 1 do
               array[n] = 0;
          end;
     end;
end;
```

Appendix G

20-Sim code of the IPC

G.1 Minimum state

```
variables
      real deltaPos, x, f, fc, deltax;
      real E, En, dEI, Eq, redE, Er, Ediss;
      real x1, x2, nx, m;
      real prevEn, preE, preve, prevx, prepos;
      boolean preflag, flag;

parameters
      real global k= 0.5;
      real global b= 0.002;
      real qPos = 9.58738e-5;
      real alpha= 1.0;

initialequations
E = 0;  preE=0; dEI=0; Eq=0; redE=0; Er=0; Ediss=0;
fc = 0;  x=0; nx =0; x1=0; x2=0; deltax = 0;
flag = false;

code
preflag= previous(flag);
prevx =previous(x);
prepos = previous(pos);
preve =previous(eCont);
preE = previous(E);
prevEn =previous(En);
redE = 0; E=0; m=0; flag=false;

deltaPos  = pos-prepos;
f = deltaPos/sampletime;

Er = fc^2*sampletime*b;
dEI = preve*deltaPos+Er;
Eq = abs(preve)*qPos;          // energy due to quantization noise
if (dEI<>0 and dEI>=Eq) then  //dissipate energy to compensate for quantization enc
    .sig.
      dEI = dEI-sign(dEI)*Eq;
end;

// avoiding numerical inacuracy by the PC104; 0 might not be 0, but a very small
    number
//if abs(deltaPos) < qPos then

if abs(deltaPos) ==0 then  //......1     no En exchange
      x=prevx;
else
```

```
        if (prevx==0) then //......2 after deadlock
            m = 0.5*k*(0.001*pos)^2;
            Ediss =Ediss-m;
            x = 0.001*pos;
        else      //...... else .....2*/
            E = prevEn - dEI;

            redE = alpha*dEI;
            if dEI<0 then        // compenstae Ediss
                if Ediss<0 then
                    if Ediss < redE then
                        E = prevEn- ((1-alpha)*dEI);
                        Ediss = Ediss - redE;
                    else
                        E = prevEn - dEI+Ediss;
                        Ediss = 0;
                    end;
                end;
            end;
            if E>=0 then    //......3 determine next x
                nx =sqrt(2*E/k);
                x1= (-prevx + nx)/(deltaPos);
                x2= (-prevx  - nx)/(deltaPos);
                if (sign(x1) == sign(x2)) then
                    if (abs(x1)<abs(x2))then
                        x=nx;
                    else
                        x=-nx;
                    end;
                else      //...... else .......**
                    if x1> 0 then
                        x = nx;
                    else
                        if x2> 0 then
                            x = -nx;
                        else
                            if x1>=x2 then
                                x = nx;
                            else
                                x = -nx;
                            end;
                        end;
                    end;
                end;
                flag = false;
            else //E<0   //...... else ......3    too much energy spent
                flag = true;
                Ediss = Ediss+E;       x=0;
            end; //......3
        end;      //......2
end;//......1

deltax = x - prevx;
fc =deltax/sampletime;

En = 0.5*k*x^2;
eCont = -k*x -b*fc;
```

...

G.2 Energy leap

```
variables
     real deltaPos, x, f, fc,deltax;
     real E, En, Ediss, dEI, Eq, redE, Er;
```

```
      real x1, x2, nx, m;
      real prevEn, preE, preve, prevx, prepos;
      boolean preflag, flag;

parameters
      real global k= 0.5;
      real global b= 0.002;
      real qPos = 9.58738e-5;
      real alpha= 1.0;

initialequations
E = 0; Ediss=0; preE=0; dEI=0; Eq=0; redE=0; Er=0;
fc = 0;  x=0; nx =0; x1=0; x2=0; deltax = 0; m=0;
flag = false;

code
preflag= previous(flag);
prevx =previous(x);
prepos = previous(pos);
preve =previous(eCont);
preE = previous(E);
prevEn =previous(En);

redE = 0; E=0; m=0; flag=false;

deltaPos  = pos-prepos;
f = deltaPos/sampletime;

Er = fc^2*sampletime*b;
dEI = preve*deltaPos+Er;
Eq = abs(preve)*qPos;          // energy due to quantization noise
if (dEI<>0 and dEI>=Eq) then  //dissipate energy to compensate for quantization enc
    .sig.
     dEI = dEI-sign(dEI)*Eq;
end;

//avoiding numerical inacuracy by the PC104; 0 might not be 0, but a very small
    number
//if abs(deltaPos) < qPos then

if abs(deltaPos)==0 then    //......1     no En exchange
      x=prevx;
else
      if (prevx==0) then //......2 after deadlock
          m = 0.5*k*(0.001*pos)^2;
           Ediss =Ediss-m;
          x = 0.001*pos;
      else      //...... else .....2*/
          E = prevEn - dEI;

          redE = alpha*dEI;
          if dEI<0 then  //compenstae Ediss
               if Ediss<0 then
                    if Ediss < redE then
                         E = prevEn- ((1-alpha)*dEI);
                         Ediss = Ediss - redE;
                    else
                         E = prevEn - dEI+Ediss;
                         Ediss = 0;
                    end;
               end;
          end;

          if E>=0 then          //......3 determine next x
               nx =sqrt(2*E/k);
               x1= (-prevx + nx)/(deltaPos);
               x2= (-prevx  - nx)/(deltaPos);
```

```
                if (sign(x1) == sign(x2)) then
                     if (abs(x1)<abs(x2))then
                          x=nx;
                     else
                          x=-nx;
                     end;
                else      //......else.......**
                     if x1> 0 then
                          if preflag then x=-nx; else x=nx; end;
                     else
                          if x2> 0 then
                            if preflag then x=nx; else x=-nx; end;
                          else
                            if x1>=x2 then
                                 if preflag then x=-nx; else x=nx; end;
                               else
                                 if preflag then x=nx; else x=-nx; end;
                               end;
                          end;
                     end;
                end;
                flag = false;
           else //E<0  //......else......3 too much energy spent
                flag = true;
                Ediss = Ediss-dEI;  x=-prevx;
           end; //......3
      end;      //......2
end;//......1

deltax = x - prevx;
fc =deltax/sampletime;

En = 0.5*k*x^2;
eCont = -k*x -b*fc;
```

Appendix H

Dimensions

The controller receives the position as input. Using this position (and thus velocity), it determines the desired torque put on the Pantograph arms;

$$e_{controller} = -k \cdot x - b \cdot f \tag{H.1}$$

in which the following dimensions apply:

$e_{controller}$: $[Nm]$
k : $[Nm/rad]$
x : $[rad]$
b : $[Nms/rad]$
f : $[rad/s]$

In the physical system, this torque is not applied to the robot arms by the controller. The output of the controller is first converted into a PWM signal (by the H-bridge), which is then put on the motors of the Pantograph. Thus, the motors apply a torque depending on the PWM signal. The PWM signal is a voltage, 24 V in this case, with a certain duty cycle that results in a net current.

The way the system is currently modelled, this duty cycle is determined by the controller; a controller output equal to 1 results in 100% PWM and a zero controller output will lead to a 0 duty cycle in the PWM signal. The controller output thus has a normalized value between -1 and 1, where the absolute value determines the duty cycle of the PWM signal. The direction in which the motor should turn is established by the sign of the controller output. This means that the parameters and variables used in the controllers are scaled to obtain an output of which the absolute value is limited to 1. Hence the controller output, representing the dury cycle ($1 = 100\%$ PWM), is dimensionless.

The relations between the controller output and the applied torque is not linear, as there is a non linearity between the H-bridge input and the current produced by the PWM signal. This relation has been assumed to be linear in the models used in this project. The controller is assumed to compensate for the error caused by this assumption.

The dimensions of the controller parameters and output can be determined using the following method; The arms driven by the motor are very light. Assuming that the motor data for no-load situation still apply when the motor is connected to a light weight set-up, the 'speed constant' and 'speed/torque gradient' can be used to approximate the quotient of torque applied by the motor and voltage put on the motor.

$$\left.\begin{array}{lcl} \text{speed constant} & : & 407\ [rpm/V] \\ \text{speed/torque gradient} & : & 40.2\ [rpm/mNm] \end{array}\right\} \Rightarrow \frac{407}{40.2} = 10.12438\ [mNm/V] \tag{H.2}$$

$$V_{H-bridge}/100\%\, PWM \;\; : \;\; 24\, V$$

Thus, by multiplying the parameters as well as the output of the implemented controller by a factor 0.24, the previously mentioned dimensions are obtained.
The obtained parameters assume a linear system, where it should be noticed that the system is not linear

Appendix I

Hardware specifications

I.1 Anything IO board

The MESA 4I65 is a general purpose programmable I/O card for the PC/104-PLUS bus [19]. Specifications:

- 200K gate Xilinx FPGA
- Several pre-made functions, including:
  - 12 channel host based servo motor controller
  - 4 or 8 channel micro-controller based servo motor controller (SoftDMC)
  - 8 channel, 32 bit timer counter card capable of running at 100 MHz
- 5V tolerant and can sink 24 mA

I.2 H-bridge

Specifications of the used H-bridge are [4]:

- Safety: current limiter; 10 A, transient over voltage protector diode (1500W)
- Control: H-bridge controller IC (Si9978) driving the four FETs
- Freerun/Braking: both possible
- Galvanic separation: max 10 Mbit
- Status LEDs for control signals
- Bidirectional LED displaying motor output (extra: turn on/off jumper)
- Logic input signals: 3.3-5V, galvanic isolated from motor supply
- 3 Safety levels:
  - active braking on software
  - active braking when Vlogic (or control signals are missing)
  - active braking on emergency switch which overrules all other options and operates on Vmotor
- Fault output signal: Detects overcurrent and/or undervoltage
- Passive cooling with heatsink mounted on top of PCB => 3.1K/W

I.3 Encoders pin-out

The following table shows the electrical connections of the encoder [22];

	R119	
Output Functions	Wire Colors Conn. Code **P**	Ribbon conn Conn. Code **Y**
A	Orange	2
/ A	Yellow	3
B	Violet	6
/ B	Gray	7
IND	Green	4
/ IND	Blue	5
+V	Red	1
COMMON	White	8
CASE		

(a) Electrical connections of the encoder

Wire color	**Conn. Pin**
Red	2
Orange	6
Yellow	5
Green	10
Blue	9
Violet	8
Gray	7
White	3

(b) Encoder conn. pin-out

I.4 Switching board

In order to interface the encoder output to the PC104 anything IO and the controller output from the anything IO board to the H-bridge, switching boards were used that were already designed for another set-up in the CE laboratory, the Production Cell [4]. Only a very small part of the functionality of this board is being used for this project. The connector configurations are shown below;

Function	**Connector PIN#**
N.C.	1
5V	2
GND	3
N.C.	4
!A	5
A	6
!B	7
B	8
!I	9
I	10

Figure I.1: Interface between the switching board and the encoder

Function	**Connector PIN#**
GND	1
Direction	2
PWM	3
Brake	4
Fault (input)	5
3.3V	6

Figure I.2: Interface between the switching board and the H-bridge

Bibliography

[1] R.J. Anderson and M.W. Spong, *Bilateral control of teleoperators with time delay*, Automatic Control, IEEE Transactions on (May 1989), no. 5, vol.34, 494–501.

[2] B. Appelhof, *Design of haptic interface technology*, Msc report, University of Twente, Enschede, The Netherlands, 2001.

[3] R. van As, *Constitutief gedrag van het basismateriaal voor een tussenwervelschijf*, (Dutch), internship report 92-065, University of Eindhoven, WFW, Eindhoven, The Netherlands, June 1992.

[4] L.S. van den Berg, *Design of a production cell setup*, MSc report 016CE2006, University of Twente, Control engineering, Enschede, The Netherlands, Juli 2006.

[5] S. Byrne, *Gizmodo*, http://www.gizmodo.com.au/2007/08/haptic_glove_transmits_feeling.html, Aug 2007.

[6] G. Campion, Qi Wang, and V. Hayward, *The pantograph Mk-II: a haptic instrument*, IEEE/RSJ International Conference on Intelligent Robots and Systems[IROS 2005] (2-6 Aug 2005), 193–198.

[7] CGI, *ClayTools and FreeForm digital 3D modeling systems*, http://www.cgindia.org/2007/05/claytools-and-freeform-digital-3d.html, May 2007.

[8] J.E. Colgate and Schenkel G.G., *Passivity of a class of sampled-data systems: Application to haptic interfaces*, Journal of Robotic Systems (1996), no. 1, vol.14, 37–47.

[9] J.E. Colgate and N. Hogan, *Robust control of dynamically interacting systems*, Journal of Robotic Systems (1988), vol.48, 65–88.

[10] cooley-dickinson hospital, *da vinci robot with surgeons*, http://www.cooley-dickinson.org/slideshows/north-building/images/38-surgeons-davinci_full.jpg, 2008.

[11] Immersion corporate, *MicroScribe 3D digitizer*, http://www.immersion.com/digitizer/support/newsletter.php?issue=2004spring, 2004.

[12] N. Deolaiti and G. Niemeyer, *Wave haptic dynamics: Providing stiff coupling to virtual environments*, IEEE symposium on haptic interfaces for virtual environment and teleoperator systems (2003), 185–192.

[13] V. Duindam, *Port-based modeling and control for efficient bipedal walking robots*, Ph.D. thesis, University of Twente, Enschede, The Netherlands, 2006.

[14] B. Hannaford and Jee-Hwan Ryu, *Time-domain passivity control of haptic interfaces*, Robotics and Automation, IEEE Transactions on (2002), no. 1, vol.18, 1–10.

[15] N. Hogan, *Controlling impedance at the man/machine interface*, Proceedings., IEEE International Conference on Robotics and Automation (14-19 May 1989), vol.3, 1626–1631.

[16] Y.S. Kim and B. Hannaford, *Some practical issues in time domain passivity control of haptic interfaces*, Proceedings. IEEE/RSJ International Conference on Intelligent Robots and Systems (2001), vol.3, 1744–1750.

[17] F.M. Klomp, *Haptic control for dummies: An introduction and analysis*, Tech. Report DCT 2006.097, Eindhoven University of Technology, Department of Mechanical Engineering, Control Systems Technology Group, Eindhoven, The Netherlands, Aug 2006.

[18] D.W. Marhefka and D.E. Orin, *A compliant contact model with nonlinear damping for simulation of robotic systems*, IEEE Transactions on Systems, Man and Cybernetics, Part A (Nov 1999), no. 6, vol.29, 566–572.

[19] Mesa, *4I65 fpga based PC104-PLUS anything I/O card*, http://www.mesanet.com, 2008.

[20] Maxon motor, *High precision drivers and systems*, catalogue, 78, datasheet 06/07, 2006.

[21] G. Niemeyer and E. Slotine, *Stable adaptive teleoperation*, International journal of robotics research (1991), no. 1, vol.16, 152–162.

[22] Gurely precision instruments, *Gurley models R119 and R120 rotary incremental mini-encoders*, datasheet v3.1, NY, U.S.A.

[23] C. Secchi, S. Stramigioli, and C. Fantuzi, *Control of interactive robotic interfaces, a port-hamiltonian approach*, Springer, Heidelberg, Germany, 2007.

[24] C. Secchi, S. Stramigioli, and C. Fantuzzi, *Delayed virtual environments: a port-hamiltonian approach*, Proceedings. IEEE/RSJ International Conference on Intelligent Robots and Systems [IROS] (27-31 Oct. 2003), vol.3, 2956–2961.

[25] E. Slotine and G. Niemeyer, *Telemanipulation with time delays*, International journal of robotics research (Sep. 2004), no. 9, vol.23, 873–890.

[26] C. Stramigioli, S. van der Schaft, A.J. Secchi, and C. Fantuzzi, *Sampled data systems passivity and discrete port-hamiltonian systems*, IEEE Transactions on Robots (2005), no. 4, vol.21, 574–587.

[27] S. Stramigioli, *Modeling and IPC control of interactive mechanical systems: A coordinate free approach*, Springer, Great Britain, 2001, lecture notes in control and information sciences 266.

[28] S. Stramigioli, C. Secchi, A.J. van der Schaft, and C. Fantuzzi, *A novel theory for sampled data system passivity*, IEEE/RSJ International Conference on Intelligent Robots and System (2002), vol.2, 1936–1941.