RESEARCHING THE EFFECTS OF TIME-DELAY ON ADMITTANCE CONTROL

E.J. Euving

FACULTY OF ENGINEERING TECHNOLOGY, BIOMECHANICAL ENGINEERING (BW)
BIOMECHATRONICA AND REHABILITATION TECHNOLOGY

EXAMINATION COMMITTEE
Prof. dr. ir. H.van der Kooij
ir. A.Q.L. Keemink
ir. J.H. Meuleman

DOCUMENT NUMBER
BW - 410

UNIVERSITY OF TWENTE.

07-08-2013

Abstract

This research was done for haptic robots used in rehabilitation of stroke victims. The goal for these haptic robots is have the least possible impedance, commonly called *zero impedance*. Users should experience the lowest possible hindrance from the robot, which in practice means that the robot should have a low sensation (virtual) mass. Admittance Control is used to achieve this goal. When the goal is pursued problems are met in the form of system instability. There are two recognized situations where instability occurs: when the robots are free-floating and when they are in contact with a constraining environment. Many variables affect system stability. A goal of designers of haptic robots is to have enough knowledge about Admittance Control that a robot can be designed and any stability problems can be predicted in the design phase. This research focussed on one of these variables: pure time-delay in the controller of Admittance Control. The effects of time-delay on stability in both free-floating as constrained environment were examined and the possibility of using a Smith predictor as a solution was explored. This was researched using analytical and numerical models for both environments as well as experiments on a haptic robot set-up for a constrained environment. The results showed that time-delay decreases stability for free-floating environments and constrained environments, even at low values. A Smith predictor greatly negates the effects of time-delay and improves Admittance Control in constrained environment, but descreases haptic quality and stability in free-floating environments. Models and experiments are in agreement, suggesting that predictive models that capture the dominant effects can be used in the design phase of haptic robots.

Contents

Chapter 1

Introduction

In 2007 around $191,000$ people suffered from a stroke in the Netherlands, with an equal occurrence among men and women. It is the third cause of death for men and second cause of death for women, with respectively $3,488$ and $5,425$ fatal strokes in 2010. Stroke survivors face other consequences of strokes. These include partial paralysis, (temporary) loss of sight or problems speaking. In the case of paralysis, rehabilitation is needed for proper recovery, though full recovery might not always be possible. Rehabilitation is time-intensive and labour-intensive, making stroke one of the most expensive deseases in the Netherlands. All figures are from the same reference [11].

In today's world robots are no longer the science-fiction they once were. Many processes today are possible because of the use of robots, either because humans will not do them, should not do them or cannot do them. A clear definition for what a robot is does not exist, but *a* definition is that it *is a mechanical or virtual agent [...] that is guided by a computer program or electronic circuitry*[16] and the Merriam-Webster dictionary states that it is *a device that automatically performs complicated often repetitive tasks*[3]. The car industry has made use of robots for years, for example, to do repetitive tasks such as welding of doors. Deep-sea exploration uses unmanned submarines, space exploration uses The Voyager space probes or area reconnaissance uses Unmanned Aerial Vehicles (UAV): these are all robots.

In different fields of medicine, robots are also used, such as the *da Vinci Surgical System*[5] which allows for surgery with minimal invasion. The surgeon controls the da Vinci without interacting directly with the patient. Likewise, in the field of rehabilitation, the Lower extremity Powered ExoSkeleton (LOPES2)[8] was developed by the University of Twente for gait training and assessment of motor function in stroke survivors.

The LOPES2 is a *defined impedance* robot. This means that it is designed to act in a specified manner when used. In layman terms, this could mean that it should feel lighter (in mass) than it really is or feel as it moves through a viscous liquid while really moving through air.

When a robot is controlled to *feel* a certain way, they can be called *haptic robots*. For example, the LOPES2 is designed and controlled to feel as light as possible; when this is a goal, the robot is also called a *zero impedance* robot. In reality an impedance of 0 is impossible, but it can be approached.

Among several forms of interaction controllers, two methods to control haptic robots are Impedance Control and Admittance Control. Take note that *defined impedance* and *Impedance Control* are separate things. Impedance Control is all about controlling the interaction forces between the user (the patient) and the robot. Admittance Control on the other hand controls the position or velocity of the robot when a user interacts with it. In other words: if an object is moved by the user, Impedance Control will control the reaction forces the user experiences. If the user exerts a force upon the object, Admittance Control will control the movement of the object.

The LOPES2 is Admittance Controlled. This is a choice (as both types are possible) which is usually more practical for heavier robots.

When a patient uses the LOPES2, their legs are clamped into the robot on three places: at the thighs, above the knee and below the knee. When using the LOPES2, two aspects are relevant to consider:

- Does the LOPES2 feel the way it should for the patient? This is haptic quality, also known as transparancy.
- Does the LOPES2 stay in control? When it becomes unstable, it will start to move with growing oscillations and the patient will no longer have control over the robot. A reason, among several, that this happens is that the patient constrains the movement of the robot and the controller is unable to cope with it.

The system's properties (negatively) affect the haptic quality and the stability. Internal time-delay is one of these properties, where time-delay is the time difference between when something is meant to happen and when it actually happens. Reasons for this include calculation times or communication times between different computers or computer parts.

This research focusses on the effects of time-delay on an Admittance Controlled robot's haptic quality and stability. A Smith predictor will be considered as a remedy to these effects. A Haptic Master™ arm (made possible by MOOG®) and a model of this set-up will used to achieve this.

Chapter 2

Fundamentals

This chapter covers some fundamental knowledge used in this research. Haptics are discussed in section 2.1, where some examples of haptic robots are given. Continuous-time systems are discussed in section 2.2, where the Laplace Transform and the Z-Transform are explained. System control is discussed in section 2.4, which includes theory about system stability.

2.1 Haptics

Haptics is any form of nonverbal communication involving touch[13]. Haptic technology (often shortened to *haptics*) is technology which takes advantage of the users' sense of touch to create virtual objects or environments. It does this using sensors to feedback information into the system.[12]

Haptics usually involves a robot interacting with the user. It is possible to have local haptics, where the robot and user are in the same place. The Haptic Master$^{\text{TM}}$ is an example of this, where the robot and user are in the same place. Alternatively, nonlocal haptics also exist. The robot isn't in the same place as the user, but a separate interface interacts with the user instead. This allows the user to work over distance. For example, surgeons operating over long distances can use haptics to aid their surgeries.[18]

2.1.1 Examples Of Haptic Robots

As mentioned before, the Haptic Master$^{\text{TM}}$ is a haptic robot. It is a 3 degrees of freedom (DOF) haptic robot which allows users to manipulate objects in a virtual environment which they see on screen (fig. 2.1). They can move the object, but more importantly, feel the object. It is also used for arm rehabilitation, as also seen in fig. 2.1.[10]

Figure 2.1: The Haptic Master™

Another aforementioned haptic robot is the LOPES2 (fig. 2.2). The purpose of the LOPES2 is help stroke victims rehabilitate. It aids in gait training by guiding patients' legs during walking and it assesses motor functions.

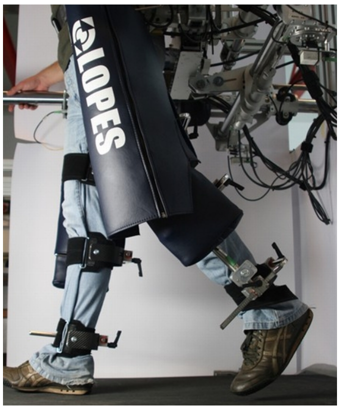


Figure 2.2: The LOPES[9], a predecessor of the LOPES2

A third example is the Simodont®Dental Trainer, produced by MOOG®(fig. 2.3). Its goal is to provide high-end dental simulation and training. Trainees can use it to practise their skills in a virtual environment where haptics ensure realism.[6]

Figure 2.3: The Simodont®

2.2 Linear Time Systems

Imagine that a certain mass-spring-damper system can be described by the following differential equation,

$$F(t) = m\frac{\mathrm{d}v(t)}{\mathrm{d}t} + cv(t) + k\int v(t)\,\mathrm{d}t \tag{2.1}$$

where $F(t)$ is the output force as a result of the input velocity $v(t)$. As long as the differential equation is linear and time invariant, the Laplace Transform is a tool to bypass these operations.

2.2.1 The Laplace Transform

The Laplace Transform is an integral transform that changes a function from one in the (continuous) time domain to one in the Laplace domain or so-called s-domain. The formal definition is as follows:[14]

$$F(s) = \mathcal{L}\{f(t)\}(s) = \int_0^\infty e^{-st} f(t)\,\mathrm{d}t \tag{2.2}$$

Where $\mathcal{L}(s)$ is the Laplace operator, $F(s)$ the Laplace transform of $f(t)$ and s is a complex number:

$$s = \sigma + i\omega, \text{ with real numbers } \sigma \text{ and } \omega.$$

The Laplace Transform has a number of properties, two of which will be mentioned.

Property 1

If

$$\mathcal{L}\{f(t)\}(s) = F(s)$$

then

$$\mathcal{L}\{\frac{\mathrm{d}f(t)}{\mathrm{d}t}\} = sF(s) - f(0) \tag{2.3}$$

For any system starting at rest, $f(0) = 0$. All systems analyzed in this report are considered to start at rest and thus $f(0)$ is never included in the calculations.

Property 2
If

$$\mathcal{L}\{f(t)\}(s) = F(s)$$

then

$$\mathcal{L}\Big\{\int_0^t f(\tau)\,\mathrm{d}\tau\Big\} = \frac{1}{s}F(s) \tag{2.4}$$

Looking at eq. 2.1 again, this is the result when it is transformed to the s-domain:

$$F(s) = msV(s) + cV(s) + k\frac{V(s)}{s} \tag{2.5}$$

Instead of an integral or differential equation, the equation is transformed to an algebraic one, making the math easier. Simple division is all that is needed to find $V(s)$:

$$\begin{aligned} F(s) &= (ms + c + k\frac{1}{s})V(s) \\ V(s) &= \frac{1}{ms + c + k\frac{1}{s}}F(s) \\ &= H_t(s)F(s) \end{aligned}$$

where

$$H_t(s) = \frac{1}{ms + c + k\frac{1}{s}}$$

$H_t(s)$ is called the *transfer function* of the system where $F(s)$ is the input and $V(s)$ the output, as shown in fig. 2.4. It describes, in the Laplace domain, the manner in which the system affects the input to produce the output. Using the Laplace transform the effect of the system (the transfer function) has become a mere multiplication.

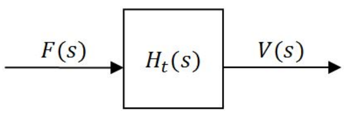


Figure 2.4: A system with transfer function $H_t(s)$

Assume that now output $V(s)$ is used as the input for another system with transfer function $H_r(s)$ and output $Y(s)$. See fig. 2.5.

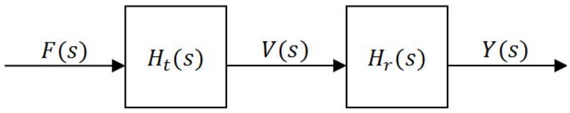


Figure 2.5: A system with transfer functions $H_t(s)$ and $H_r(s)$

Due to the fact that everything is done in the Laplace domain, the output $Y(s)$ can be described as

$$\begin{aligned} Y(s) &= H_r(s)V(s) \\ &= H_r(s)H_t(s)F(s) \\ &= H_{total}(s)F(s) \end{aligned}$$

where

$$H_{total}(s) = H_r(s)H_t(s)$$

If $v(t)$ or $y(t)$ are wanted (in the time domain), take the inverse Laplace Transform of $V(s)$ or $Y(s)$ respectively.

In short, the Laplace Transform is a tool that makes the analysis of interconnected systems easier when these systems are described by linear differential equations.

If an input or system isn't in the continuous time domain, but rather in the discrete time domain, there is a tool very similar to the Laplace Transform to make system analysis easier: the Z-transform.

2.3 The Z-Transform

What Laplace Transform is to continuous time systems, the Z-Transform is to discrete time systems. As each experiment was done in the discrete-time domain, all used Laplace Transforms were converted to Z-Transforms. The formal definition is as follows:[17]

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n]z^{-n} \tag{2.6}$$

and in the case of a causal system (a system that can exist in the real world),

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=0}^{\infty} x[n]z^{-n} \tag{2.7}$$

where $\mathcal{Z}$ is the Z-Transform operator, $X(z)$ is the Z-Transform of $x[n]$ and z is, similar to s, a complex number. The Z-Transform has an important property called time shifting which states that if

$$\mathcal{Z}\{x[n]\} = X(z)$$

then

$$\mathcal{Z}\{x[n-k]\} = z^{-k}X(z) \tag{2.8}$$

An Example

Imagine the following discrete system with output $y[n]$ and input $x[n]]$ where

$$y[n] = x[n] + 2x[n-1] + 4x[n-2]$$

Using the time shifting properly, the Z-Transform is

$$\begin{aligned} Y(z) &= X(z) + 2z^{-1}X(z) + 4z^{-2}X(z) \\ &= (1 + 2z^{-1} + 4z^{-2})X(z) \end{aligned}$$

and the transfer function

$$H(z) = \frac{Y(z)}{X(z)} = 1 + 2z^{-1} + 4z^{-2}$$

This example neatly shows that time-delays (such as taking input $x[n-1]$ instead of $x[n]$) become z^{-k} in the Z-domain, where k is the delay in samples.

2.4 System Control

In fig. 2.2.1 a mass-spring-damper system was described and the Laplace Transform was explained to gain insight on the system. When that system should be controlled (i.e. the output should be controlled) then *System Control* or *Control Theory* become relevant.

2.4.1 Control Theory

Control Theory is a field of engineering and mathematics that deals with the behaviour of dynamics systems. If the output $V(s)$ of the aforementioned system should follow a certain reference $V_{ref}(s)$, then Control Theory makes this possible. This is usually done with a controller and some form of feedback, as seen in the generalized scematic in fig. 2.6.

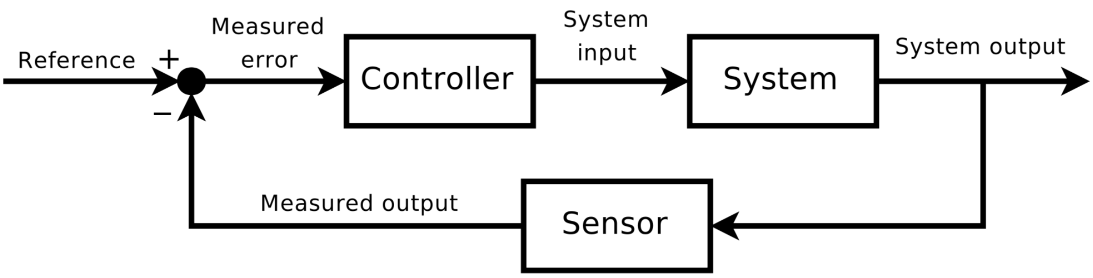


Figure 2.6: Generic Control Theory [Wikipedia]

The closed loop transfer function $H_c(s)$ for the total system seen in fig. 2.6, with *reference* as input and *System output* as output and the total system everything between these two, is

$$H_c(s) = \frac{C(s)S(s)}{1 + C(s)S(s)E(s)} \tag{2.9}$$

with controller $C(s)$, system $S(s)$ and sensor $E(s)$.

2.4.2 System Stability

Any system can become unstable, either because it is inherently unstable or the controller turns it unstable. The system in fig. 2.6 is unstable if the real parts of the poles of $H_c(s)$ are positive. A pole is any value of s (in the Laplace domain) which causes the denominator $D(s)$ of the transfer function to become zero:[15]

$$H(s) = \frac{N(s)}{D(s)}$$

Any value of s which makes

$$D(s) = 0$$

is a pole of $H(s)$ and a root of $D(s)$. $N(s)$ and $D(s)$ are polynomials. Finding the poles of a transfer function is a method to discover instability. When the real parts of the poles of $H(s)$ are positive, the solution of the system described by the relevant differential equation increases out of bounds.

Another method uses the Nyquist stability criterion. This is a graphical technique to determine whether a system is stable. It will not show *what* the poles of the transfer function are, but it will show *how many* unstable poles there are in the closed loop system. For large and complex transfer functions, the Nyquist stability criterion can show where or not it is stable.

The Nyquist criterion looks at the characteristic equation $1 + C(s)S(s)E(s)$ of $H_c(s)$ rather than looking at the whole closed loop transfer function. $C(s)S(s)E(s)$ is plotted in a complex plane for all s. If $C(s)S(s)E(s) = -1$, the denominator of $H_c(s)$ is zero and the system is unstable. If the graph encircles the real number -1 clockwise on the Nyquist plot, the system is marginally stable. If the graph goes around the left of the real number -1 (in clockwise direction) the system is unstable.

Control theory makes it possible to design and control (haptic) robots.[19]

Chapter 3

Admittance Control

In this chapter the basics of Admittance Control will be described and explained. This will be done step by step, scematics to show the different systems.
Using Control Theory, haptic robots interact with users in such a way that they *feel* different; the experienced (virtual) device or environment is different from the real robot or environment. These robots can manage this through Impedance or Admittance Control.[1]

The main idea behind Impedance Control[4] is to measure some displacement due to an external force which is then fed through a mathematical model: the impedance model. This gives the responsive forces the system should then apply to the environment or human to feel like some object. If the user of an Impedance Controlled robot exerts a force against it, the Impedance Control will control the reaction forces.

The main idea behind Admittance Control is to measure some interaction force between the robot and the user, upon when the movement (velocity) of the robot is controlled. The measured force is fed through a mathematical model: the admittance model. This model relates force to velocity. A controller will regulate the velocity of the robot according to the experienced forces. In this manner the robot will also feel like some object by the way it moves. It does not have to be limited to velocity control, position control or acceleration control are also possible. The choice depends on the requirements of the situation.

This research will focus on Admittance Control.

The systems that are discussed will be considered linear and thus noise will be fully ignored. Sensors are usually a source of noise, but they are considered noise-less. Likewise, it is assumed that quantization does not play a role.
In haptic robots, the admittance model can be (but is not at all limited to linear systems) the transfer function $H_a(s)$ of a mass-spring-damper system with virtual inertia m_v, the virtual stiffness k_v and damping c_v of a device or environment, as seen in eq. 3.1. All

calculations are done in the Laplace domain for reasons described in section 2.2.1.

$$H_a(s) = \frac{V(s)}{F(s)} = \frac{1}{m_v s + c_v s + \frac{k_v}{s}} \tag{3.1}$$

For a given force, the admittance model would give a certain velocity. Schematically shown in fig. 3.1, with input force F, output velocity $\dot{x}$ and transfer function H_a. Take note that everything is done in the Laplace domain and '(s)' has been left out for simplicity.

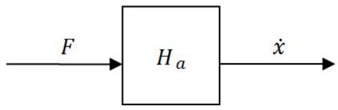


Figure 3.1: Admittance Control, scematic 1

The given velocity can be considered the reference velocity. This is the velocity which the (moving parts of the) robot must achieve. It will be called the plant from now on. For the plant P to move as it should, a controller C must regulate the current I or voltage (in most electrical devices it is one of these two quantities) going into the plant to get it moving. See fig. 3.2.

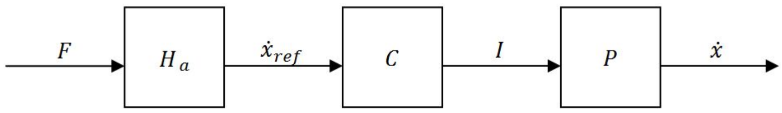


Figure 3.2: Admittance Control, scematic 2

In a perfectly known world with an ideal controller, $\dot{x}_{ref}$ and $\dot{x}$ are the same. It is safe to assume that neither the world nor the controller is ideal. For this reason feedback is usually included into the Admittance Control, as seen in fig. 3.3. e is the error; the difference between the reference velocity $\dot{x}_{ref}$ and output velocity $\dot{x}$.

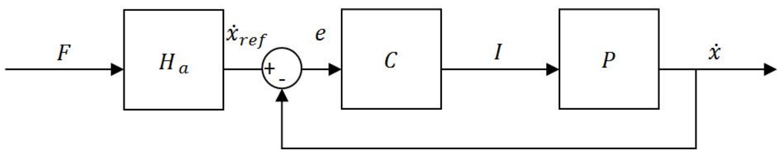


Figure 3.3: Admittance Control, scematic 3

A feedback controller C_{fb} now controls the plant according to the error in velocity it encounters. Still, it is also safe to assume some knowledge about the plant. If the robot, mechanically speaking, behaves like a mass on a spring it will work very similar to transfer function H_a, albeit with different parameters. Using this knowledge, a feedforward controller C_{ff} can be added to the schematic to supply known necessary forces. This feedforward controller will work similar to the ideal controller in fig. 3.2, with the feedback controller able to correct any errors. See fig. 3.4.

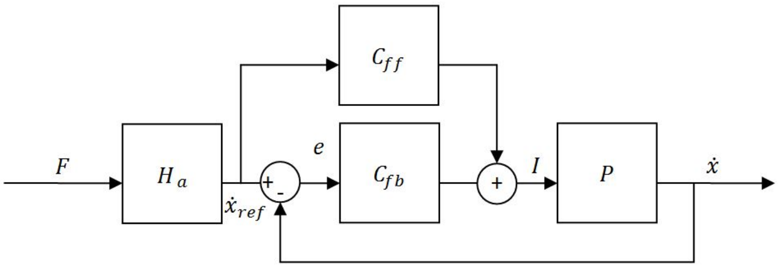


Figure 3.4: Admittance Control, scematic 4

Up till now it hasn't been discussed where the given forces for the admittance model come from, but to properly control a haptic robot they must be identified as well as possible. In the current situation, the forces acting on the robot are measured, thus the force sensors are part of the plant. External sources, such as a human interacting with the robot, can be considered first. Next, forces created by the plant must be taken into account. A haptic robot will use force sensors to measure forces.

Ideal force sensors measure forces without affecting the system or plant, but in reality there are things to take into account. Most notably, post-sensor masses, damping effects and finite sensor stiffness affect the measured forces. Figure 3.5 shows a generic hardware set-up for an Admittance Controlled robot. There are no arrows, as the direction of processes hasn't been specified. The human interacting with the robot is part of the *Environment Mechanics* and will affect the sensed force. In addition to these environment mechanics there are *Post-Sensor Mechanics*. For instance, if there is a mass attached to the outer end of the force sensor, it will generate inertial forces during motion. These will be added to the sensed force and now the Admittance Control will compensate for that too.

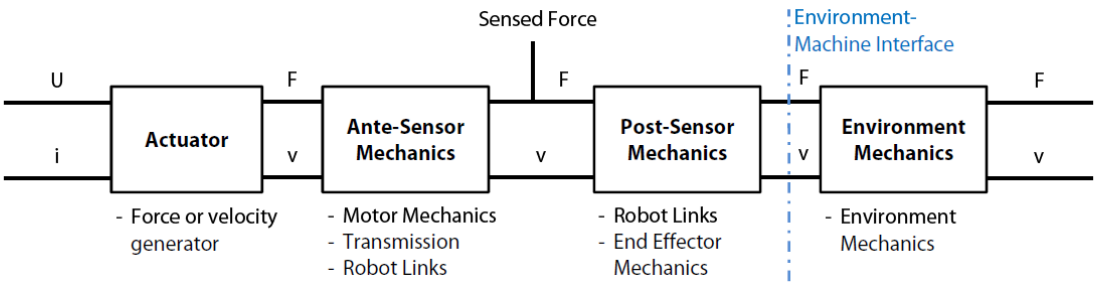


Figure 3.5: Generic Admittance Control Hardware

As the plant moves, the sensors will measure forces due to these effects and these enter the system again. Q represents the transfer function for these post-sensor effects in fig. 3.6. Note that the forces also directly affect the plant.

A logical next step is to identify the sensor and more specifically, the post-sensor effects which create the additionally measured forces. If these are properly known they can be cancelled using a negating model. Calling this R, it can be added into the schematic seen in fig. 3.6.

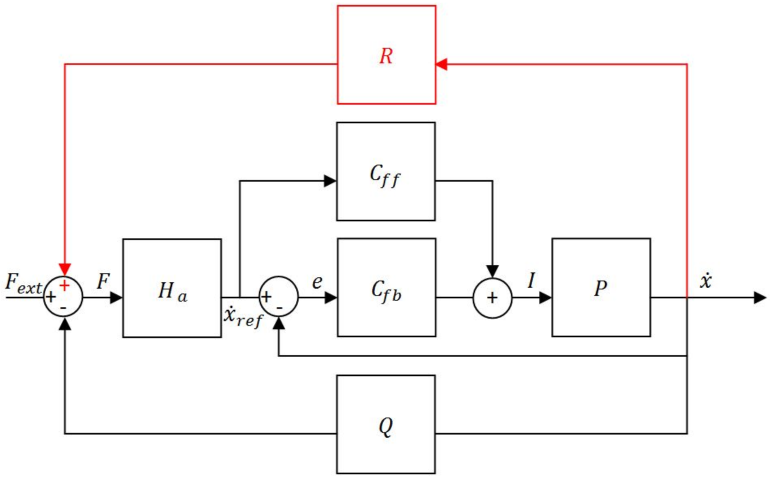


Figure 3.6: Admittance Control, scematic 5

Finally the quality of the sensors should be taken into account. A perfect sensor has a transfer function of 1 at all frequencies. This means that the input signal equals the output signal. At higher frequencies, the sensor can no longer 'keep up' and the result is phase shifting (the output signal starts lagging behind) and lower amplutides. In reality, a sensor usually has a bandwidth where it functions ideally but once the frequency (for example) goes outside the bandwidth of the sensor, the transfer function is no longer 1. In the set-up that has been schematically shown before, two quantities need sensors: force and velocity. The force sensor S_f and velocity sensor $S_{\dot{x}}$ are added to the scematics in fig. 3.7.

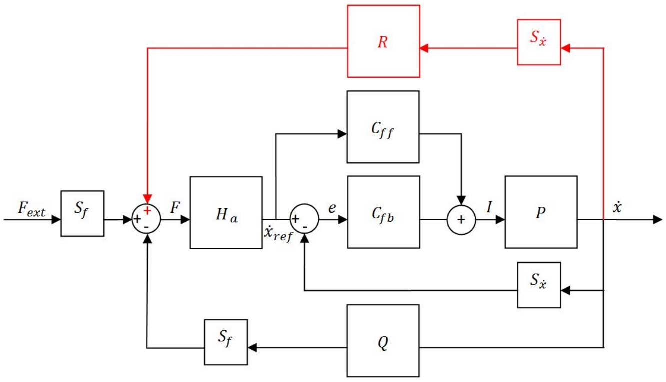


Figure 3.7: Admittance Control, scematic 6

The last schematic is an example of one for a robot with Admittance Control. If these were added to the scematics, each sensor signal would first be quantized and then noise would be add to that signal. In the next chapter the schematic in fig. 3.7 will be expanded to show the control for the Haptic Master^{TM} arm used in this bachelor assignment.

Chapter 4

The Situation

As mentioned in section 2.4.2, an Admittance Controlled system can become unstable. To get a clear understanding of where the instability in the set-up with the Haptic Master™ comes from, it is necessary to have a better idea of the situation and the control involved with the robot.
In this chapter the theoretical model (section 4.1), time-delay and the Smith predictor are explained. The theoretical model will be based on the set-up and the stability of the Admittance Control will be examined. Two definitions to measure the stability are given: haptic quality and constrained stability. In section 4.2 the effects of time-delay are discussed from a theoretical perspective and in section 4.3 the Smith predictor is explained in discussed.

4.1 Theoretical Model

The admittance control for the set-up using the Haptic Master arm is schematically shown in fig. 4.1. The forward path from F_{ext} towards the plant P is the result of the fact that all external forces (F_{ext}) also affect the plant directly.

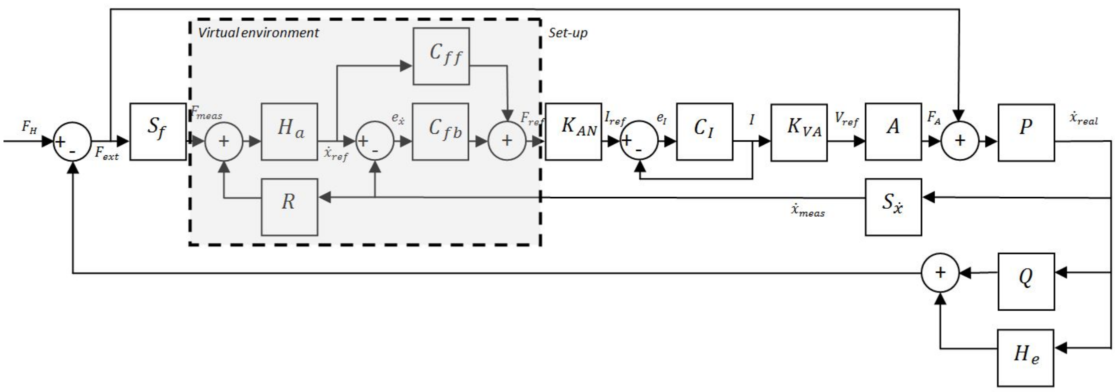


Figure 4.1: Admittance Control, full scematic

Details, from top to bottom and left to right:

- C_{ff} Feedforward Controller
- F_h Force, Human
- F_{ext} Force, External
- S_f Sensor, Force
- F_{meas} Force, Measured
- H_a Admittance Model, chosen to have only inertia.
- $\dot{x}_{ref}$ Velocity, Reference
- $e_{\dot{x}}$ Error, Velocity
- C_{fb} Feedback Controller
- F_{ref} Force, Reference
- K_{AN} Ratio, Ampre per Newton
- I_{ref} Current, Reference
- C_I Current Controller
- e_I Error, Current
- K_{VA} Ratio, Volt per Ampre
- V_{ref} Voltage, Reference
- A Actuator, the motor which recieves a voltage and produce a force (indirectly via torque and a gear ratio)
- F_A Force, Actuator
- P Plant, the moving part of the robot which produces a linear velocity
- $\dot{x}_{real}$ Velocity, Real
- R Post-Sensor Effects, Model
- $\dot{x}_{meas}$ Velocity, Measured
- $S_{\dot{x}}$ Sensor, Velocity
- Q Post-Sensor Effects, for example due to post-sensor mass
- H_e Environmental Impedance, for example due to contrains

The dashed box indicates the parts which are done in a virtual environment (Simulink, in this case) with everything else being in a physical environment. Some things that are important to take note of:

- The actuator produces a purely rotational force, better known as torque. A gearing ratio relates the torque to the linear force, which also assumes neglectable losses due to finite stiffness of the leadscrew (more on this in chapter 3). For the sake of simplicity this ratio is considered part of the actuator, as if it produces linear forces.
- The force sensor measures linear forces but outputs a voltage. For the sake of simplicity the volt-per-newton ratio is considered part of the sensor, as if it outputs the measured forces.
- The velocity sensor measures rotational velocity of the leadscrew, rather than linear velocity. Similarly as with the actuator, the gearing ratio relates the rotational velocity with the linear velocity. For the sake of simplicity this ratio is considered part of the sensor, as if it outputs the measured linear velocity

4.1.1 Stability Of Admittance Control

Generally, feedback loops are sources for instability. If fig. 4.1 is simplified, the loops become obvious. For this simplification the feedforward controller $C_{ff} = 0$, the sensors and actuator are ideal and the forces acting upon the robot do not influence the movement (the forward path of F_{ext} going to the plant is taken out). Lastly, the ratios K_{AN} and K_{VA} is taken out. See fig. 4.2.

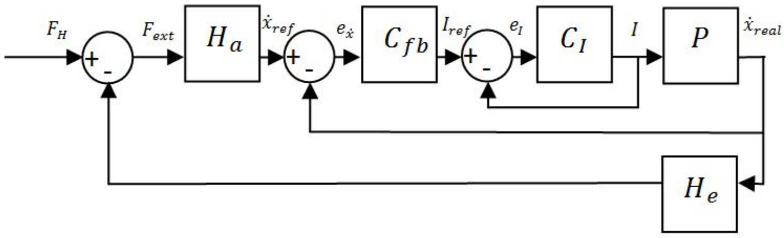


Figure 4.2: Admittance Control, simplified

The transfer function for each loop be described as follows:

$$H_I = \frac{I}{I_{ref}} = \frac{C_I}{1 + C_I} \tag{4.1}$$

$$H_C = \frac{\dot{x}_{real}}{\dot{x}_{ref}} = \frac{C_{fb}H_IP}{1 + C_{fb}H_IP} \tag{4.2}$$

$$H_t = \frac{\dot{x}_{real}}{F_H} = \frac{H_aH_C}{1 + H_aH_CH_e} \tag{4.3}$$

Using Nyquist criterion it is possible to quickly determine which loop creates any instability, once the variables are filled in.

It should be mentioned that a fourth loop could be recognized, namely the post-sensor dynamics compensation loop, where the post-sensor dynamics model R compensates for the post-sensor dynamics Q. The assumption is that R properly negates the effects of Q and thus can be ignored.

In this research it is also assumed that H_I will not be a source of instability. The current controller is commercially available and assumed to be stable. This case the transfer function H_t becomes

$$H_t = \frac{H_a H_C}{1 + H_a H_C H_e} \tag{4.4}$$

where

$$H_C = \frac{C_{fb} P}{1 + C_{fb} P} \tag{4.5}$$

Using arbitrairy numbers it can be shown that H_e can make the system unstable. For that,

- $P = \frac{1}{m_P s + c_P}$ if potential stifness and time-delay of the plant are ignored.
- $C_{fb} = \frac{k_p s + k_i}{s}$ if a PI controller is chosen.
- $H_a = \frac{1}{m_v s}$ if only a virtual mass is wanted.
- $H_e = c_e + k_e s$ if there is an environment with damping and stiffness.

where

- $m_P = 0.1\mathrm{kg}$
- $c_P = 0.25\frac{\mathrm{Ns}}{\mathrm{m}}$
- $k_p = 83\frac{\mathrm{Ns}}{\mathrm{m}}$
- $k_i = 4.5\frac{\mathrm{N}}{\mathrm{m}}$
- $m_v = 0.5\mathrm{kg}$
- $c_e = 0.01\frac{\mathrm{Ns}}{\mathrm{m}}$
- k_e will be varied from $0\frac{\mathrm{N}}{\mathrm{m}}$ to $4\frac{\mathrm{N}}{\mathrm{m}}$ to show that at certain stiffness values the system becomes unstable.

Figure 4.3 shows the Nyquist plots of the system with different environmental stiffness values. The environment had to have *some* damping to ever be stable at all, thus the $c_e = 0.01\frac{\text{Ns}}{\text{m}}$. With this damping, the system is stable for $k_e = 0\frac{\text{N}}{\text{m}}$ which has no real parts. It is also stable for $k_e = 1\frac{\text{N}}{\text{m}}$, as can be seen by the fact that it intersects itself *left* of -1 (depicted by a red $+$). The red line, for $k_e = 2\frac{\text{N}}{\text{m}}$ is critically stable as it passes right through -1. The other two lines, for $k_e = 3\frac{\text{N}}{\text{m}}$ and $k_e = 4\frac{\text{N}}{\text{m}}$, both intersect themselves *right* of -1 which indicates instability in these plots.

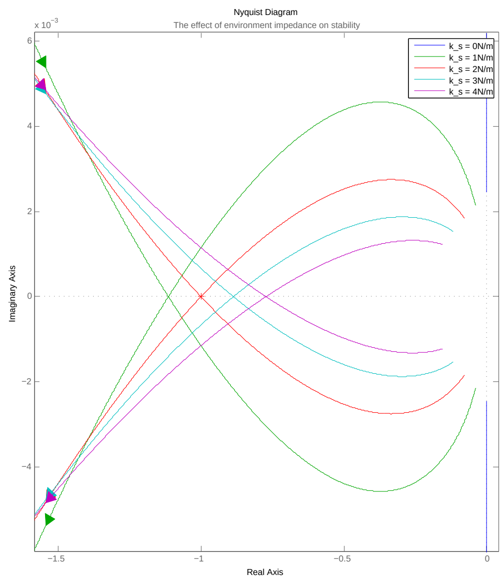


Figure 4.3: Nyquist plot of the system with different environmental stiffness.

In order to access stability, some measure of quality is needed. Two such measures are defined for this assignment: *haptic quality* and *constrained stability.*

Haptic Quality

Haptic quality ϵ is defined as

$$\epsilon = \int |\ln(H_a(s)) - \ln(H_t(s))|^2 \, \mathrm{d}s \tag{4.6}$$

In most cases $s = j\omega$ and there is a certain relevant bandwidth, which means that the definition can be rewritten into the following equation:

$$\epsilon = \int_{\omega_{min}}^{\omega_{max}} |\ln(H_a(j\omega)) - \ln(H_t(j\omega))|^2 \, \mathrm{d}\omega \tag{4.7}$$

Constrained Stability

When an unconstrained robot (free-floating) hits a wall or a sheet of metal, it suddenly experiences constrains from that wall or sheet. Within milliseconds the Admittance Control reacts to correct the movement of the robot, but due to properties of the environment (high stiffness, for example) it can overcompensate and react excessively, causing instability. In this research the environment will be considered merely stiff ($H_e = \frac{k}{s}$) as this is the most practical to experiment with.

If the environment acts as a spring, the robot can be made to collide with it. The test will measure the incoming and outgoing velocity of the robot. The ratio between the two is defined as the constrained stability

$$\eta = \frac{\dot{x}_{outgoing}}{\dot{x}_{incoming}} \tag{4.8}$$

4.1.2 Analysis

Now, the Haptic Master arm will be perfectly admittance controlled if the total transfer function,

$$H_t = \frac{\dot{x}_{real}}{F_H} = H_a \tag{4.9}$$

To gain some (analytical) insight on when this is the case, H_t has to be calculated first. This is done by starting at the output ($\dot{x}_{real}$) and working back towards the input(F_H).

$$\begin{aligned}
\dot{x}_{real} &= P(F_A + F_{ext}) \\
F_{ext} &= F_H - (H_e + Q)\dot{x}_{real} \\
\dot{x}_{real} &= P(F_A + F_H - (H_e + Q)\dot{x}_{real}) \\
F_A &= K_{VA} A I \\
I &= \frac{C_I}{1 + C_I} I_{ref} \\
K_I &= \frac{C_I}{1 + C_I} \\
I_{ref} &= F_{ref} K_{AN} \\
F_{ref} &= C_{ff}\dot{x}_{ref} + C_{fb}(\dot{x}_{ref} - S_{\dot{x}}\dot{x}_{real}) \\
\dot{x}_{ref} &= H_a(S_f F_{ext} + R S_{\dot{x}}\dot{x}_{real})
\end{aligned}$$

There are no more new variables in the last equation and thus it's possible to start filling in the unknowns to create the transfer function,

$$H_t = \frac{\dot{x}_{real}}{F_H} = \frac{P(H_a\kappa(C_{ff}+C_{fb})S_f+1)}{P(H_a\kappa(((H_e+Q)S_f - RS_{\dot{x}})(C_{ff}+C_{fb})+C_{fb}S_{\dot{x}})+H_e+Q)+1} \tag{4.10}$$

where

$$\kappa = K_{VA}AK_{AN}K_I$$

This is quite a large function, but it can be simplified. For the time being, a few assumptions are made:

Assumption 1
The sensors are ideal. Thus, $S_f = S_{\dot{x}} = 1$

Assumption 2
The bandwidth of the current controller is higher than that of the feedback controller. Thus, $K_I = 1$

Assumption 3
The actuator properly applies the reference force F_{ref}. Thus, $\kappa = 1$. This assumption is based on assumption 2.

Using these assumptions, the new transfer function becomes,

$$H_t = \frac{P(H_a(C_{ff}+C_{fb})+1)}{P(H_a(((H_e+Q)-R)(C_{ff}+C_{fb})+C_{fb})+H_e+Q)+1} \tag{4.11}$$

The question which was to be answered was, 'when is the robot perfectly admittance controlled?', i.e. $H_t = H_a$. Consider a few idealizations.

Idealization 1

Firstly, the post-sensor dynamics Q can be cancelled if they are properly identified and modelled. In such a case $Q = R$ and the transfer function becomes slightly smaller,

$$H_t = \frac{P(H_a(C_{ff}+C_{fb})+1)}{P(H_a(H_e(C_{ff}+C_{fb})+C_{fb})+H_e+Q)+1} \tag{4.12}$$

Idealization 2

If the feedback controller C_{fb} was perfect, it would react instantaneously. Also, the feedforward controller C_{ff} would no longer be needed. Thus, $C_{fb} \to \infty, C_{ff} = 0$ and,

$$\lim_{C_{fb}\to\infty} H_t = \frac{H_a}{1+H_aH_e} \tag{4.13}$$

Figure 4.4 shows a scematic of idealization 2 and eq. 4.13 would indeed be the transfer function of this system.

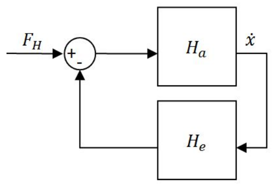


Figure 4.4: Idealization 2, scematically

If the environment H_e has no constrains ($H_e = 0$) then

$$H_t \approx H_a$$

This is exactly what is wanted.

4.2 Time-Delay

In section 4.1.2 everything was done under the assumption that the Admittance Control was done time-continuous. In reality, in the case of this research it is done time-discretely at a sampling frequency of 1000Hz. As a result, the idealization in section 4.1.2 is only partially valid because the feedback controller won't react instantaneously. Naturally, the question whether it would compensate properly (even if it *did* react instantaneously) should be considered too.

The result is instability. Any 'mistakes' the feedback controller makes can cause phase shifts which in turn can lead to positive feedback. The small error in velocity becomes bigger and soon the Admittance Control is unstable.

One of the reasons instability occurs is pure time-delay. When a controller measures its input and adjusts its output accordingly, it loses time. Note that this is different from settling time, as illustrated by fig. 4.5.

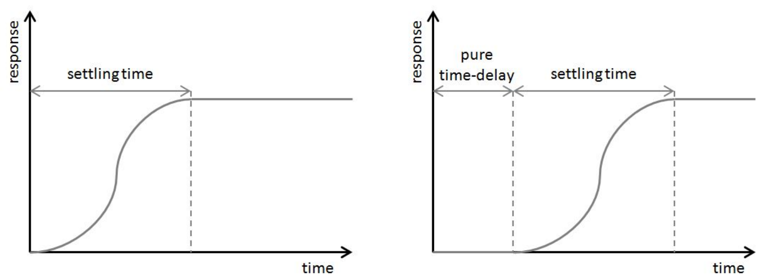


Figure 4.5: Settling time versus pure time-delay

At least one time sample passes between reading the current input (such as velocity) and producing a new output (such as a reference force). The fact that in reality Admittance Control isn't time-continuous (something which is assumed in chapter 2) means that by definition there will be time-delay. The manner in which this affects performance and stability of a local Admittance Controlled haptic system is unknown, but time-delay causes phase shifts which in turn lead to instability.

The purpose of the research is to increase the knowledge about time-delay and instability in Admittance Control in general. The goal is therefore to document how time-delay affects the set-up. Ignoring the fact that the current system already has some time-delay (this was determined to be around 1.25ms) in it, artificial time-delay will be added to the Admittance Control and its effects on stability will be measured. The effects will be measured using the given defitions of haptic quality and constrained stability. A potential solution to the time-delay is the Smith predictor, which requires knowledge of the controller and plant.

If time-delay is included into the scematics seen in chapter 2, it would be found after the controllers and the set-up:

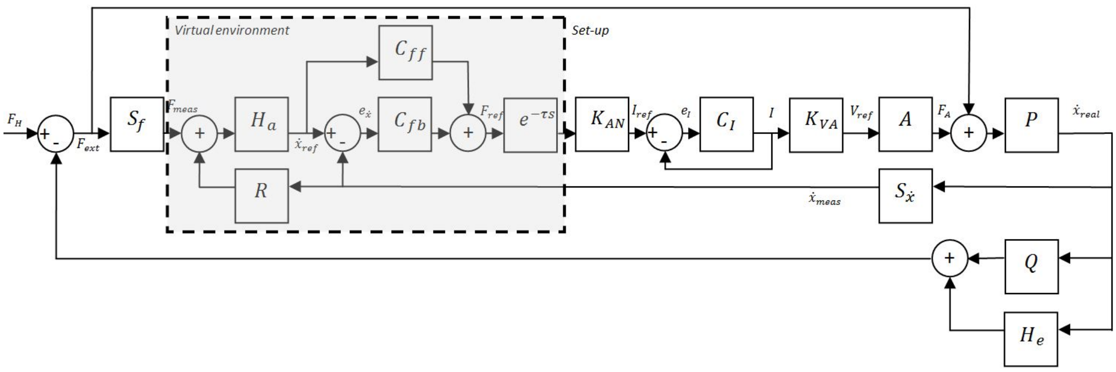


Figure 4.6: Admittance Control, full scematic with time-delay

The new transfer function H_t would become

$$H_t = \frac{P(\kappa(C_{ff} + C_{fb})H_a S_f e^{-\tau s} + 1)}{P(H_a \kappa(((H_e + Q)S_f - RS_{\dot{x}})(C_{ff} + C_{fb}) + C_{fb} S_{\dot{x}})e^{-\tau s} + H_e + Q) + 1} \tag{4.14}$$

where

$$\kappa = K_{VA} A K_{AN} K_I$$

If $\tau = 0$ then H_t would return to the form found in eq. 4.10. Next the effect of time-delay on stability is analyzed. For that, the following assumptions are made:

Assumption 1

The sensors are ideal. Thus, $S_f = S_{\dot{x}} = 1$

Assumption 2
The actuator properly applies the reference force F_{ref}. Thus, $\kappa = 1$.

Assumption 3
There is no feedforward controller used. Thus $C_{ff} = 0$.

Assumption 4
There is no environmental impedance H_e. Thus $H_e = 0$.

Assumption 5
Post-sensor dynamics are compensated. Thus $R = Q$.

With that in mind,

$$H_t = \frac{P(C_{fb} H_a e^{-\tau s} + 1)}{P(C_{fb} e^{-\tau s} + Q) + 1} \tag{4.15}$$

and individually,

- $P = \frac{1}{m_P s + c_P}$ if potential stifness and time-delay of the plant are ignored.
- $C_{fb} = \frac{k_p s + k_i}{s}$ if a PI controller is chosen.
- $H_a = \frac{1}{m_v s}$ if only a virtual mass is wanted.
- $Q = m_{ps} s$ if there is only post-sensor mass.

which produces:

$$H_t = \frac{\frac{1}{m_P s + c_P}\left(\frac{(k_p s + k_i) e^{-\tau s}}{m_v s^2} + 1\right)}{\frac{1}{m_P s + c_P}\left(\frac{(k_p s + k_i) e^{-\tau s}}{m_v s^2} + m_{ps} s\right) + 1} \tag{4.16}$$

with

$$D_t = \frac{1}{m_P s + c_P}\left(\frac{(k_p s + k_i) e^{-\tau s}}{m_v s^2} + m_{ps} s\right) + 1$$

which can be rewritten as

$$D_t = \frac{(m_P + m_{ps}) m_v s^3 + c_P m_v s^2 + k_p s e^{-\tau s} + k_i e^{-\tau s}}{(m_P s + c_P) m_v s^2} \tag{4.17}$$

Finding the roots of D_t analytically is impossible and thus it is faster to use a numerical approach. Choosing arbitrary values of correct order for each constant, it becomes possible to make Nyquist plots of different time-delays. The chosen constants:

- $m_P = 0.1$kg

- $c_P = 0.25\frac{\text{Ns}}{\text{m}}$
- $k_p = 83\frac{\text{Ns}}{\text{m}}$
- $k_i = 4300\frac{\text{N}}{\text{m}}$
- $m_v = 0.5\text{kg}$
- $m_{ps} = 0.1\text{kg}$

Figure 4.7 shows the results of Nyquist plots of H_t computed with $\tau = 0\text{ms}$,$\tau = 3\text{ms}$ and $\tau = 6\text{ms}$

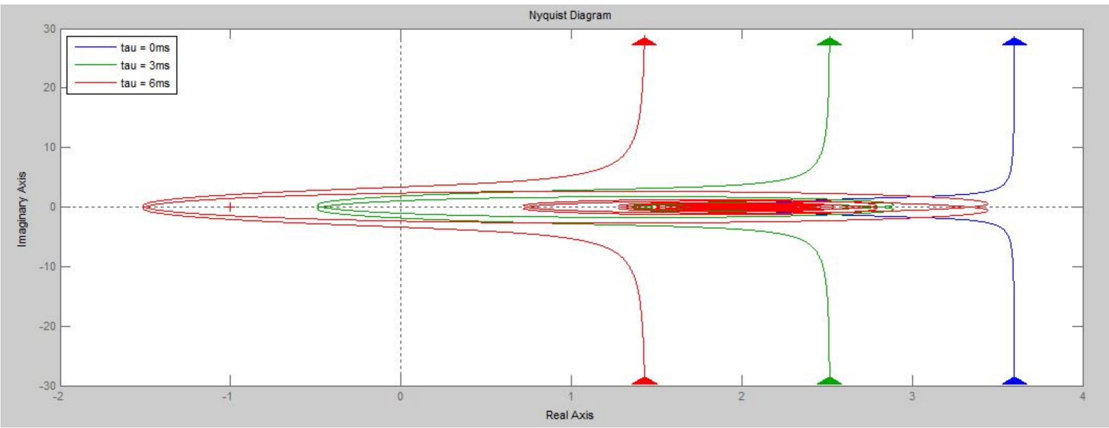


Figure 4.7: The effects of time-delay on a system

Between $\tau = 3ms$ and $\tau = 6ms$ the system becomes unstable, as seen in fig. 4.7.

4.3 Smith Predictor

A solution to time-delay is the Smith predictor. It is a predictive controller which is added to the existing controller to counter time-delay. Knowledge of the time-delay and the plant is required to have an effective Smith predictor.

4.3.1 General Theory

Suppose a system with time-delay as shown in fig. 4.8 with transfer function

$$H = \frac{Ce^{-\tau s}P}{1 + Ce^{-\tau s}P} \tag{4.18}$$

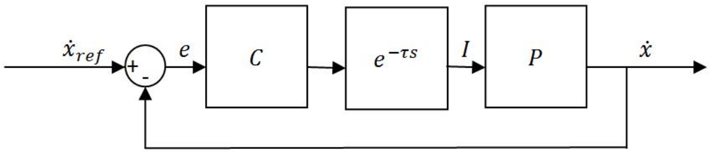


Figure 4.8: A generic system with time-delay

As shown earlier, time-delay will decrease the system's stability. The idea behind the Smith predictor is to design a Smith controller $\tilde{C}_{smith}$ such that

$$\tilde{H}_{smith} = \frac{\tilde{C}_{smith} e^{-\tau s} P}{1 + \tilde{C}_{smith} e^{-\tau s} P} = e^{-\tau s} \frac{CP}{1 + CP} \tag{4.19}$$

If eq. 4.19 is solved

$$\tilde{C}_{smith} = \frac{C}{1 + CP(1 - e^{-\tau s})} \tag{4.20}$$

The P in eq. 4.20 is renamed $\tilde{P}$ as it's a model of the real plant. Similarly, $\tilde{\tau}$ is an approximation of τ. The system with Smith predictor is shown in fig. 4.9.

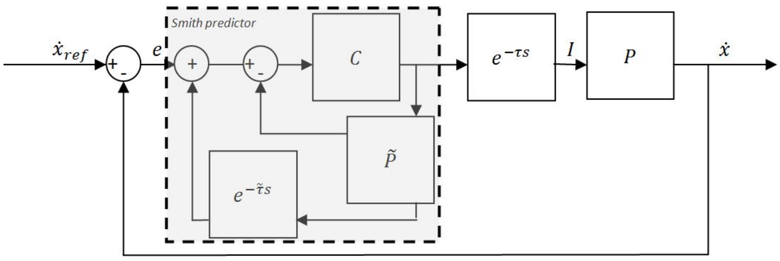


Figure 4.9: A generic system with time-delay and Smith predictor

4.3.2 Design

Figure 4.6 shows the full scematic of the Haptic Master^{TM}. When designing the Smith predictor, the same assumptions were used as in section 4.2 in addition to one more: the forward path of F_{ext} going to P was ignored.

The reason the assumptions, including the last one regarding the forward path, are deemed acceptable is due to the Smith predictor's purpose. It is designed to negate the time-delay the controller experiences. That is its purpose and thus the focus lies purely on the time-delay, controller and the plant. The effects which aren't directly involved aren't considered, as seen with the assumptions. Lastly, if the forward path was included into the analysis, the Smith predictor would have to use information from the future to function (rather than using $e^{-\tau s}$, it would use $e^{\tau s}$). This is impossible.

The transfer function for which a Smith predictor is designed is as follows:

$$H_t = \frac{PC_{fb}H_a e^{-\tau s}}{1 + PC_{fb}e^{-\tau s}} \tag{4.21}$$

With a Smith predictor $\tilde{C}_{fb}$ the new transfer function should be

$$\tilde{H}_t = \frac{P\tilde{C}_{fb}H_a e^{-\tau s}}{1 + P\tilde{C}_{fb}e^{-\tau s}} = e^{-\tau s}\frac{PC_{fb}H_a}{1 + PC_{fb}} \tag{4.22}$$

which, when solved, obtains

$$\tilde{C}_{fb} = \frac{C_{fb}}{1 + C_{fb}P(1 - e^{-\tau s})} \tag{4.23}$$

This is the same as $\tilde{C}_{smith}$ obtained for the generic system and thus the scematics will be the same as shown in fig. 4.9. The Smith predictor $\tilde{C}_{fb}$ will be used as part of the full system shown in fig. 4.6. It will be part of H_t seen in eq. 4.14, where it will substitute C_{fb}.

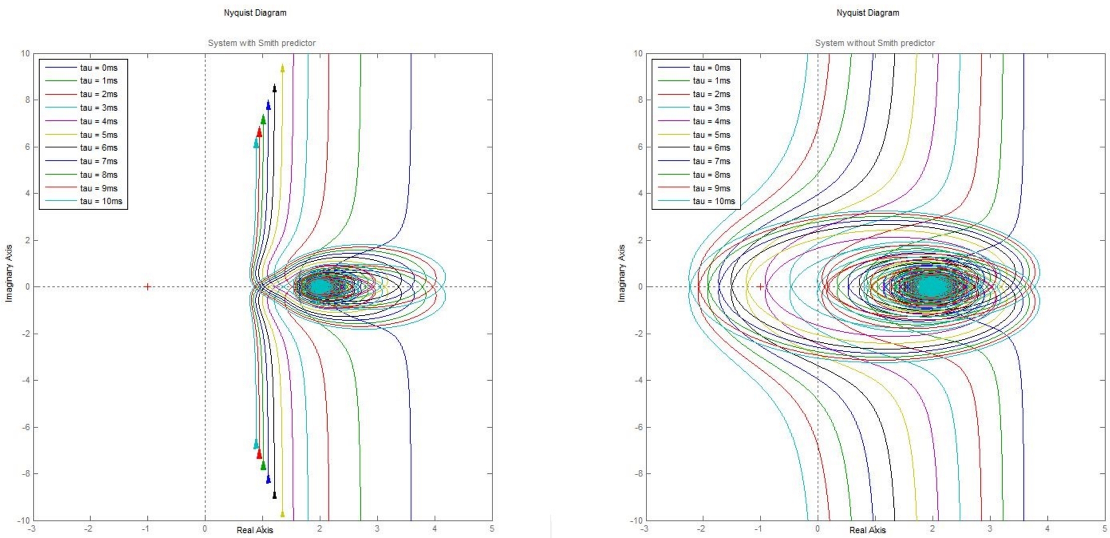
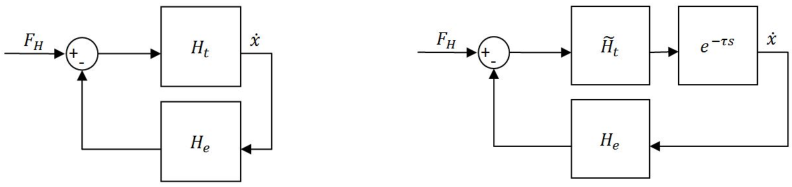


Figure 4.10: Nyquist plots with and without Smith predictor

Figure 4.10 shows the effect of Smith predictor on the stability of the system. The two Nyquist plots show eleven lines, ten with some form a time-delay and one without. The red '+' marks the '-1' point and the lines that turn clockwise around that point are unstable systems. In this situation, the system normally becomes unstable between τ = 4ms and τ = 5ms. With the Smith predictor, none the current time-delays make the system unstable. The left plot shows the results with Smith predictor, the right plot the results without Smith predictor.

The Smith predictor promises to stabilize the system, but it will only affect the haptic quality (where instability is considered to be bad haptic quality). The Smith predictor was designed for a free-floating situation where the environment impedance $H_e = 0$. Thus, when the environment starts constraining the system, the time-delay will still affect its stability. This is shown in fig. 4.11.



(a) without Smith predictor (b) with Smith predictor

Figure 4.11: The system with a environment impedance

Figure 4.11a shows the situation before introduction of a Smith predictor, where time-delay is included in H_t. Once the Smith predictor is introduced, $\tilde{H}_t$ is the new transfer function for the system and it has no more time-delay. Instead, the time-delay has been 'taken out' and follows after $\tilde{H}_t$. This is shown in fig. 4.11b. This loop is very similar to the loop shown in fig. 4.8 and thus instability due to time-delay has not been negated with the Smith predictor.

Chapter 5

The Method

The purpose of this chapter is to describe the method used to find the effects of time-delay on the used set-up and to test the effectiveness of the Smith predictor.

To start experimenting with the set-up, a number of steps had to be taken. Those steps are described below.

1. A properly working Simulink model was made to control the Haptic Master$^{\mathrm{TM}}$ arm. To avoid further confusion, this will be known as the *Control Software*. The Admittance Model, the Admittance Control and different forms of safety precautions are part of the Control Software.

2. The plant was identified. The plant was everything that was not part of the Control Software. I.e. the set-up was plant.

3. A feedback PI-controller[7] was designed for the identified plant.

4. Using the results from the identification of the plant, a numerical model was made to recreate reality. This model was made in Simulink and used to test all experiments before they were tried in reality. This is known as *the model* from now on.

5. Artificial time-delay was added to both the Control Software as the model to experiment with the effect of it.

6. The designed Smith predictor was added to the Control Software and the model to experiment with the effect of it. The time-delay used in the Smith predictor was the same one as imposed on the system via the deliberate time-delay. The purpose of the experiments was not to identify time-delay, it was to find out the effect of a working Smith predictor on the system and the manner in which it counters time-delays.

Once everything was ready two types of experiments were done, one for the haptic quality and one for the constraineed stability.

To discover the haptic quality $\epsilon(s)$ of the Admittance Control for a certain time-delay τ, a known input signal was given to the force sensor to simulate forces upon the sensor. The movement of the robot was then measured and from these results a transfer function would be obtained. This transfer function is $H_t(s)$ of eq. 4.6. Time-delay would be varied, but also the virtual mass m_s of admittance model H_a. After all, H_a is part of the outer loop transfer function H_t (eq. 4.3) found in section 4.1.1.

To discover the constrained stability η of the Admittance Control, a aluminium wall was added to the Haptic Master$^{\mathrm{TM}}$ arm against which the robot (and force sensor) collided. The incoming and outgoing velocities were be recorded at different time-delays and virtual masses.

Chapter 6

The Execution

This chapter discusses the execution of the method described in the previous chapter. A number of steps had to be taken before experimenting could start on the set-up and these involved making Control Software and the numerical model. These two are described in section 6.2 and 6.3 respectively. The set-up itself is described earlier in section 6.1.

6.1 The Set-Up

During the experiments a dismantled Haptic Master arm (as seen in fig. 6.1b, from now on referred to as *arm*) was used as frame. It was used upside down and the part that normally moves was mounted to the table, leaving the support free to move. This support, recognizable as an aluminium block in fig. 6.1b, will be referred to as *block* from now on.

(a) The set-up (b) Haptic Master arm

Figure 6.1: The set-up

Figure 6.1a and fig. 6.2 show the final set-up, which will be further explained below.

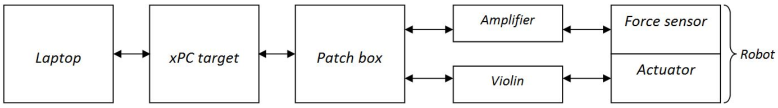


Figure 6.2: The Set-Up, scematically

The block had one degree of freedom, where the direction away from the motor (Parvex RS130ER1100) was defined as positive and the direction toward the motor as negative. A nut connected the block to a lead screw which ran through the length of the arm. In this manner circular motion from the motor was converted to linear motion, by some gearing ratio. The actuator converted current into circular torque, which created the circular motion. A motor constant related the torque to the current.

During the experiments, an xPC target computer connected the motor to the (virtual) controller. Between the actuator and the xPC target a Violin[2] was placed to control the current passing through the motor. It does this by monitoring the voltage over the motor, which is proportional to the current by Ohms law. The Violin was the current controller referred to in fig. 4.1.

As the block moved up and down the lead screw, two sensors kept track of the position and velocity of the block. These were the encoder (C2 encoder, part of the motor) and the tachometer (TBN103 tacho, part of the motor), both of which were part of the motor. The tachometer worked inversely to the actuator where circular motion created a voltage; it was a dynamo.

Fixed upon the block was a 3-D force sensor (a design used on the Simodont$^{\mathrm{TM}}$ and Haptic Master$^{\mathrm{TM}}$,as seen in figures 6.1a and 6.3). Only the forces in parallel direction were measured. The forces were measured using strain gauges which would produce varying voltages. An amplifier (SG-3016) amplified these voltages before they were sent to the patch box.

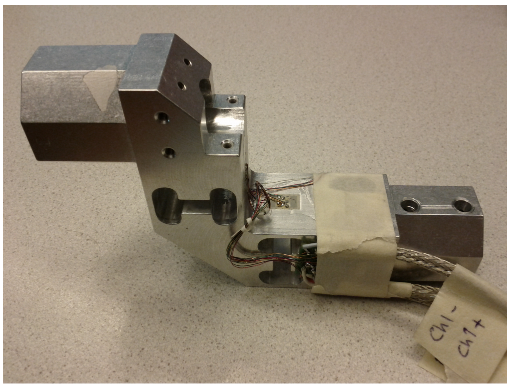


Figure 6.3: 3-D force sensor

6.2 The Control Software

For anything to happen, Control Software was needed to control the plant. A number of things were included into the Control Software that were considered needed or wanted.

Admittance Control
: The purpose of the Control Software was to have an Admittance Controlled plant. This meant it would have to have an Admittance Model and velocity controller.

Calibration
: The Control Software had to calibrate itself each time it ran to minimize errors. The possibility to calibrate mid-run was also wanted.

Safety
: Anything that could be done within the Control Software to ensure safe experiments was considered wanted. For instance, if excessive velocities or currents were measured, the plant would be turned off. If the block hit the sides or the force sensor measured extreme forces the Control Software would enter the safety state, which would shut down the plant.

These three are further explained in the upcoming subsections, with the expection of the velocity controller. That will explained in another section. Before the Admittance Control could function properly, the system had to calibrate. In addition to the position, velocity and force sensors, the motor also had to be calibrated. The borders of the arm were specified and off-sets accounted for.

6.2.1 Calibration - Position

The first step was to calibrate the encoder. This was the first state: by sending a constant negative force (strictly this is a current, but relations are linear) to the motor, the block moved to the negative direction. Once the block reached the negative edge of the arm, it would stop. The encoder wouldn't measure a variance in position and the system knew it had reached the negative side. The encoder count was recorded and given the name '`Pos_NegBorder`'.

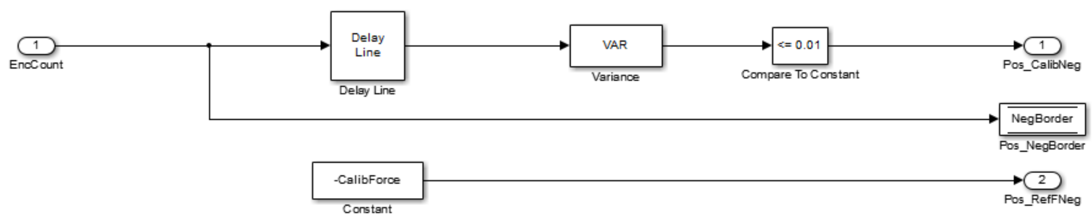


Figure 6.4: Calibration Position

Calibrating the positive border was done in the same manner, albeit in opposite direction. A constant force was sent to the motor, the block moved and once it reached the positive edge, the encoder count was given the name '`Pos_PosBorder`'.

6.2.2 Calibration - Velocity

Calibrating the tachometer happened during the calibration of the encoder. In states 1 and 3 the block moved in the negative and positive direction, respectively. While the block was at the negative edge and at rest (the variance of the tachometer voltage would be next to zero now), the voltage from the tachometer was measured and recorded. This was given the name '`Tacho_NegBorder`' and done in state 2.

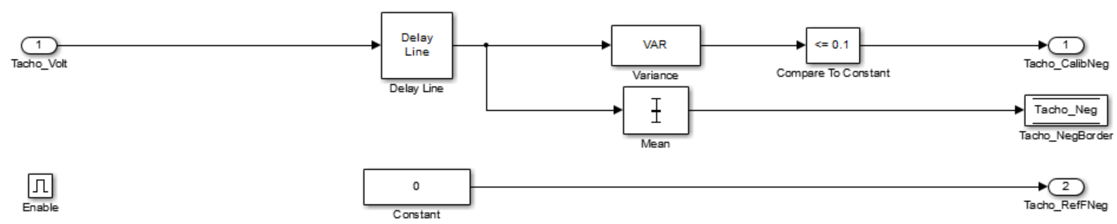


Figure 6.5: Calibration Velocity

After state 3, during which the block moved to the positive edge, the calibration was repeated. Now the measured voltage from the tachometer was given the name '`Tacho_PosBorder`'. This was done in state 4.

By taking the average of `Tacho_NegBorder` and `Tacho_PosBorder`, the offset of the tachometer was known and could be corrected. Now the tachometer was ready for use.

6.2.3 Calibration - Force

The force sensor was calibrated after the encoder and tachometer. This was state 5. The block was at the positive edge and at rest (the variance of the force sensor would be next to zero) so the voltage from the force sensor could be measured. This was recorded and given the name '`Force_PosBorder`'. Assuming that the force sensor offset wasnt position dependent, the value of `Force_PosBorder` was the offset of the force sensor. This could be corrected and the force sensor was ready for use.

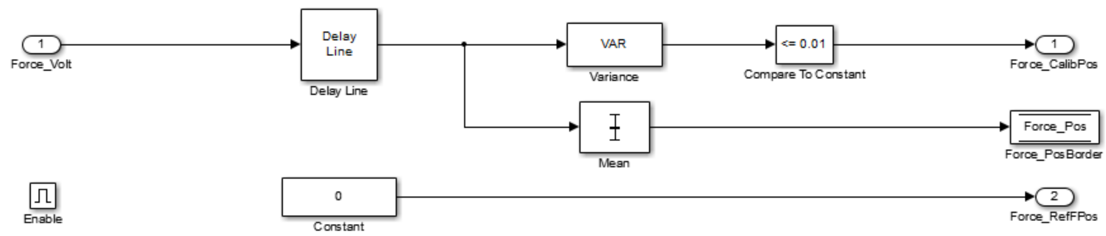


Figure 6.6: Calibration Force

6.2.4 Calibration - Motor

When the block was at the positive edge and at rest, the variance in the current going to the motor would be next to zero. In state 6, the current going to the motor was measured and recorded. This was given the name '`Current_PosBorder`' and considered the offset of the motor. This could be corrected and the motor was ready for use.

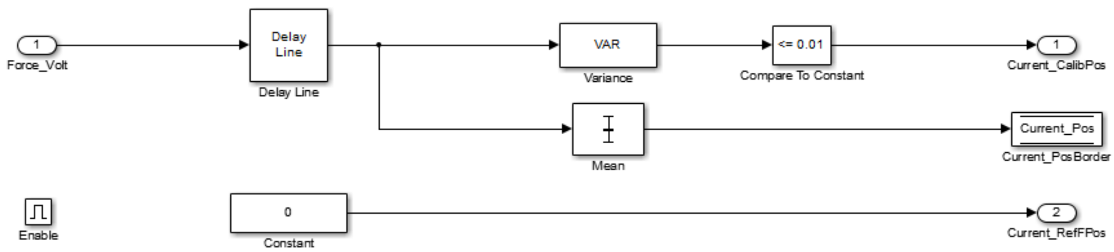


Figure 6.7: Calibration Motor

6.2.5 Calibration - Homing

The last step which was taken before the controller is ready for use is called 'homing'. This isn't calibration in the strictest sense, but it prepared the set-up for use. Thus it is still considered calibration in this report. This was done in state 7. Here the block moved to the middle of the arm, using the information gained from calibrating the encoder, tachometer and motor. With `Pos_NegBorder` and `Pos_PosBorder` it was possible to pinpoint where the middle was. Future encoder counts were relative from this middle which was given the name '`EncoderCount_Mid`'.

While the block wasn't at the reference position (the value of the middle, which is 0) there was an error. Using a simple PI velocity controller, the block was sent to the middle. Once the middle was reached, the variance in `EncoderCount_Mid` would drop and become next to the zero. The system could now go the next state, ending the calibration process.

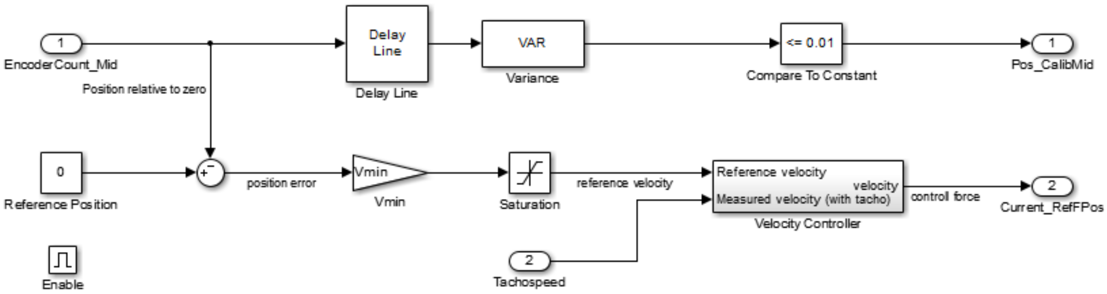


Figure 6.8: Calibration Homing

6.2.6 Velocity Control

Velocity control used a reference velocity to produce a reference force for the motor. If the controller was a feedback controller, it would use the measured velocity to compare the current velocity to the reference velocity and make changes to the reference force accordingly. The feedforward controller assumed knowledge of the system to predict what changes would have to be made and it predicted the correct reference force. The output (reference force) ensured that the block moved with the reference velocity.

During the experiments a PI controller was chosen as feedback controller with a proportional gain $k_p = 83$ and integral gain $k_i = 4.3$ for a bandwidth of 10Hz. No feedforward controller was used, even if it is deceptively shown in fig. 6.9.

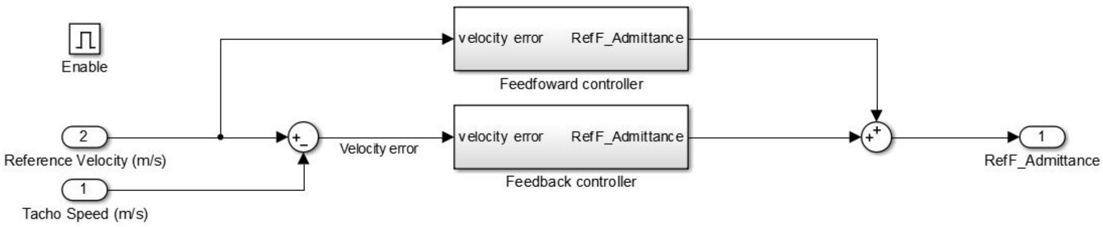


Figure 6.9: Velocity Control

6.2.7 Admittance Model

At the core of the Control Software lies the admittance model. The model linked the wanted virtual mass to the real applied force, creating the haptic environment that was wanted. The input force was the one the sensor measures. Dividing this force by the virtual mass gave the needed acceleration. With a discrete integrator this became the reference velocity.

As a safety measure, virtual borders were added to the system and these created a distance between the final end position of the block and the actual edge of the arm. The current position was compared to the two reference positions (negative and positive) and if needed, the force used by the admittance model was changed to zero. See [appendix 2] for the function.

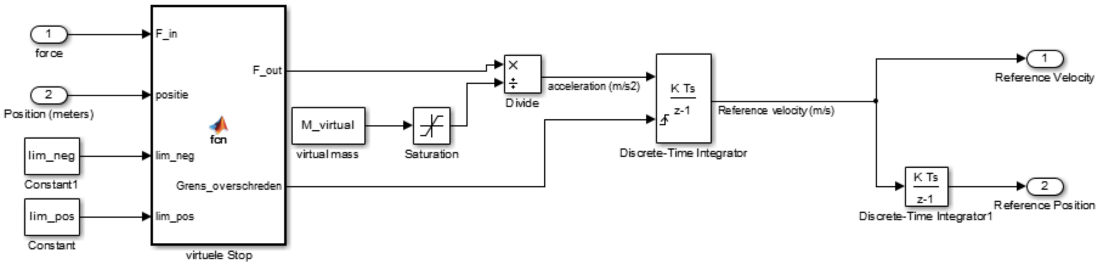


Figure 6.10: Admittance Model

6.2.8 Motor Control

All movement of the block was facilitated by the motor which was controlled by the Violin. The violin controlled the current passing through the motor by monitoring the voltage over the motor. The relation between current passing through the motor and the force acting on the block was linearly proportional and simple gain related the current to the force. The Control Software worked with a force loop, thus the motor was asked to act forces upon the block. With the gearing constant [m/rad] and the motor constant [Nm/A] it was possible to calculate the ratio between current and force in A/N. Next, using the properties from the Violin the ratio between potential difference and force was calculated in V/N. This was the output that left the Control Software. The $+1$ and -1 gains seen in fig. 6.11 meant that the signal was sent twice to the Violin, once positively and once negatively. This allowed the Violin to take the difference between these two signals to discover the actual signal. The advantage of this is that it negated most of the noise or any voltage peaks (as that difference won't change).

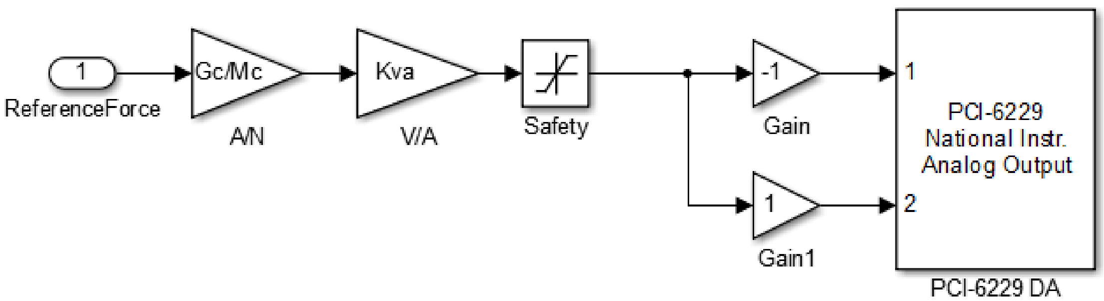


Figure 6.11: Motor Control

6.3 The Numerical Model

The following section looks at the (numerical) model. Important aspects of the model are the plant, the noise (of the sensors), the post-sensor dynamics and the stiffness of the environment. These aspects have impact on the experiments in reality and needed to be as correct as possible for the model. Each aspect has its own section where its identification is explained.

6.3.1 Identification Of The Plant

The purpose of identifying the plant was to find a transfer function that described the plant. A model could then be made with which reality could be recreated. An altered version of the Control Software was used to identify the plant. The Admittance model and velocity controller were taken out and replaced by an input source. This source sent a reference force directly to the plant. The changes in velocity were measured and processed. A transfer function was then fitted to the results.

The signal that was used as input was a series of sine waves at different frequencies. This made it easier to process the results because each set of data corresponded with a certain frequency. The range of frequencies was chosen to go from 0.1Hz to 30Hz in 15 steps. The selected frequencies were logarithmically spaced as the Bode plots would also be logarithmically spaced. In addition to different frequencies, different amplitudes were chosen, ranging from 2.5N to 6.5N with 0.5N differences. The reason to try multiple amplitudes was to discover when the transfer function of the plant could be considered linear.

Figure 6.12 shows the results of the identification of the plant For each run, the input was *reference force* and the output was *measured velocity*. The results were processed into four plots. Going from left to right, top to bottom, the first plot shows a Bode magnitude plot of the results. This magnitude plot was created by analysing the results in the fourier domain. The second plot shows another Bode magnitude plot of the same results, but this time it was created by analyzing the results in the time domain. This was possible because the input waves had set frequencies at set times. This second magnitude plot was used to verify the first one as though the methods differed, the magnitude plots shouldn't have. The third plot is the Bode phase plot, also produced by analysis in the frequency domain. The fourth plot shows the Total Harmonic Distrotion (THD) of each run, per frequency. The equation used to calculate the THD was

$$\mathrm{THD} = \frac{\sqrt{\sum\limits_{n=2}^{m} A_n^2}}{A_1} \tag{6.1}$$

where

$m =$ highest harmonic that is still below the Nyquist frequency

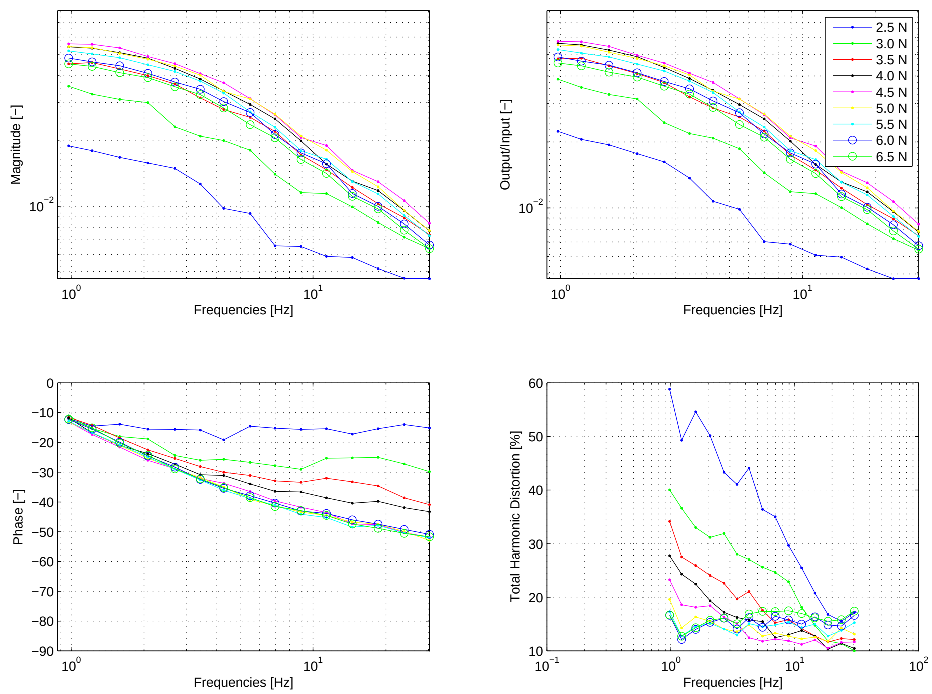


Figure 6.12: Identifying the plant

The purpose of doing THD analysis on the results was to determine what weight a data point could have. The higher the THD, the lower the weight of the point and the less it was taken into account when fitting transfer functions to the results.

Using Matlab's `lsqnonlin` function for nonlinear data-fitting a transfer function was fitted to each of the results, taking their weights into account. In order to find a reasonable fit, the form of the transfer function had to be defined. It was assumed that the plant would have inertia m_p, damping c_p and some time-delay τ_p, making the transfer function of the following form:

$$H_p(s) = \frac{1}{m_p s + c_p} e^{-\tau_p s} \tag{6.2}$$

The fit for the amplitude of 5N is shown in fig. 6.13, as that is the fit that was used to define the plant. It is added to the first and third plot, indicated by dashed lines of equal colours. The transfer function H_p for the plant was thus found to be

$$H_p(s) = \frac{1}{0.6028s + 22.9652} \tag{6.3}$$

where

$$m_p = 0.6028\text{kg} \tag{6.4}$$

$$c_p = 22.9652\frac{\text{Ns}}{\text{m}} \tag{6.5}$$

$$\tau_p = 0\text{ms} \tag{6.6}$$

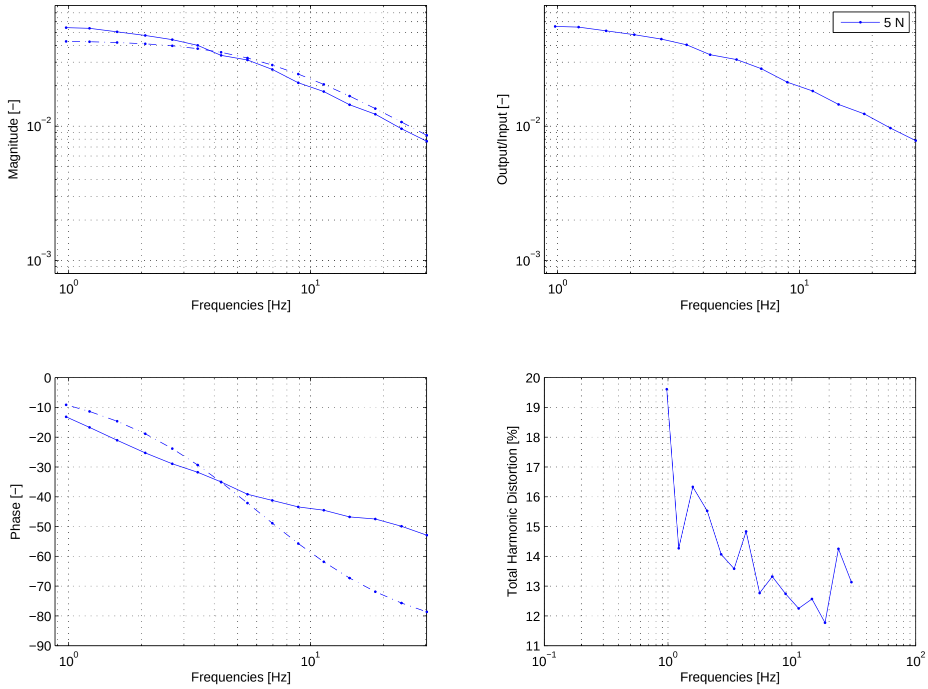


Figure 6.13: Identifying the plant (note the different frequency axis in the THD plot)

The fit does not fully match the phase plot, but match with the magnitude plots is good. This was considered more important.

6.3.2 Noise

Preliminary tests showed that the tachometer and the force sensor experienced noise. This would affect the Admittance Control during experiments and thus it had to be included into the model. The identification for the noise was done rather simply: each sensor was first calibrated and then left to measure 'nothing' for 10 minutes. The data measured during that time was considered noise and that data was added to model.

6.3.3 Post-Sensor Dynamics

To identify the post-sensor dynamics the same data was used that was used to identify the plant. Instead of using measured velocity as output, it was used as input and the measured force was the output, since the movement of the robot and the post-sensor dynamics would cause a force to be measured. The fit was done expecting only post-sensor mass, so the transfer function had the form

$$H_{ps}(s) = \frac{1}{m_p s} \tag{6.7}$$

Figure 6.14 shows the result of the signal with an amplitude of 5N and its fit, resulting in

$$m_{ps} = 0.088\text{kg} \tag{6.8}$$

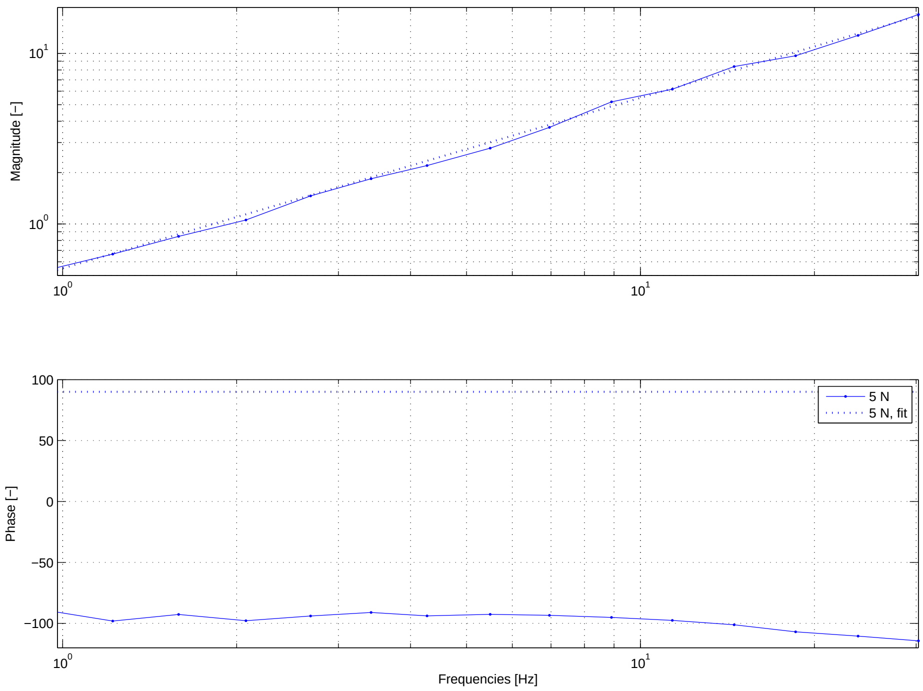


Figure 6.14: Identifying the post-sensor dynamics

6.3.4 Stiffness Of Environment

The stiffness of the aluminium wall was identified by measuring the forces as the robot acted upon it. Figure 6.15 shows the results. The displacement is from the middle of the arm, but the wall only started at 0.07m, thus there is an offset on the fit function. The gradient is relevant, though, as the magnitude of it indicates the stiffness of the block. In addition, the data isn't very clean and prone to errors, so the stiffness won't be very correct. The order of the stiffness (10^5) could be taken from this, though, and thus

$$k_e = 10^5 \frac{\text{N}}{\text{m}} \tag{6.9}$$

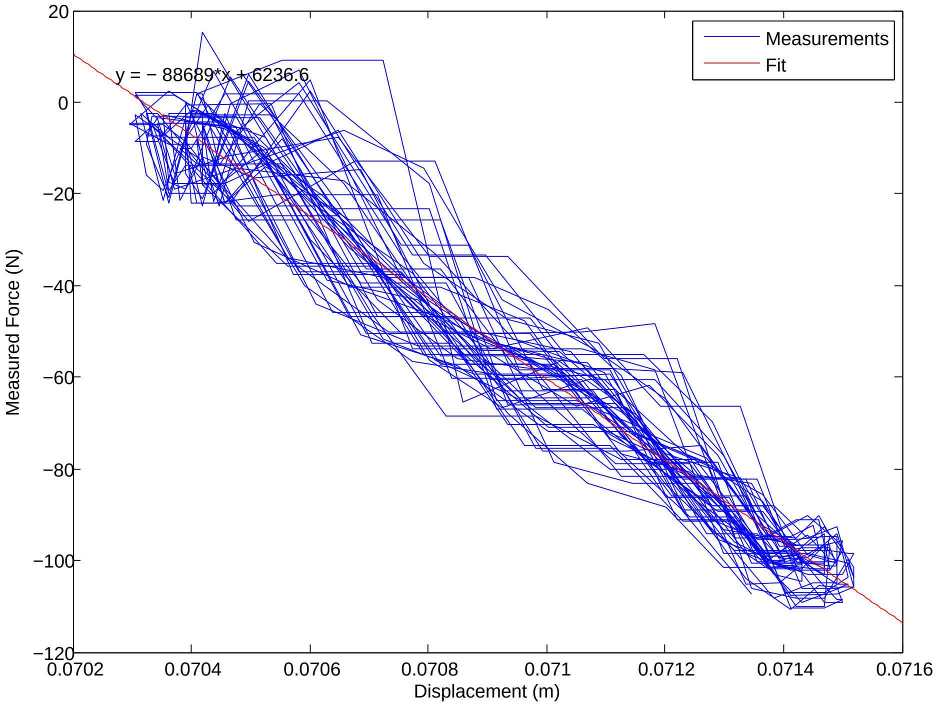


Figure 6.15: Identifying the stiffness of the environment

Chapter 7

Results

This chapter shows the results from the two series of experiments to discover the haptic quality (section 7.1) and constrained stability (section 7.2) of Admittance Control with time-delay and with the addition of a Smith predictor.

7.1 Haptic Quality

Discovering the haptic quality of the set-up turned out be to unfeasible. It turned out that the noise on the force sensor had enough effect to disrupt each experiment, regardless of chosen time-delay or virtual mass. The noise would either move the block's position out of bounds or make the system unstable enough to increase the velocity untill it too went out of bounds. After numerous tries to fix the issues with noise it was decided to discontinue experiments with the set-up. Experiments with the model were done.

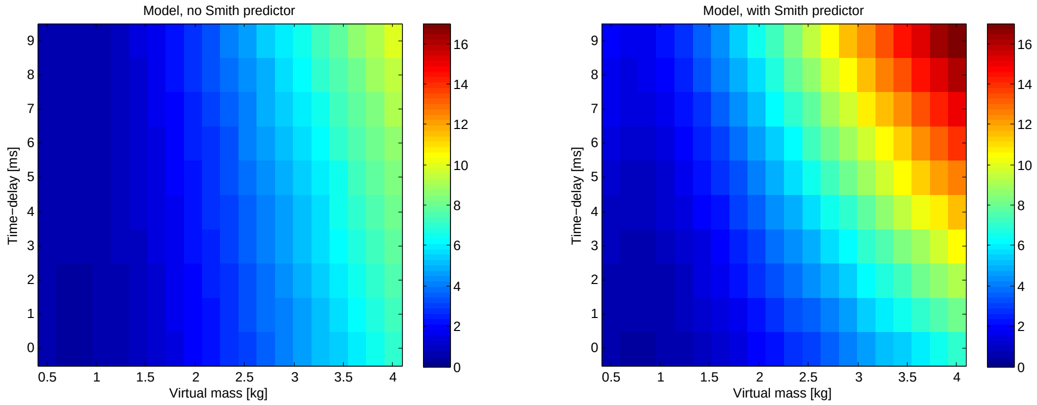
7.1.1 The Model

For these experiments, multi-sine signal was sent to the (virtual) force sensor with an amplitude of 10N. The signal lasted for a 100 seconds and had a total of 20 frequencies between 0.1Hz and 10Hz. These frequencies were spread out logarithmically. The measured velocity was then used to calculate the transfer function H_t. This H_t was compared to

$$H_a = \frac{1}{m_v s}$$

using eq 4.7. These gave haptic quality ϵ.

Figure 7.1 shows the results with and without the Smith predictor. The colour depict the values for ϵ.



(a) without Smith predictor

(b) with Smith predictor

Figure 7.1: Haptic quality (the colours depict the values) of the Admittance Control of the model with and without the Smith predictor

7.2 Constrained Stability

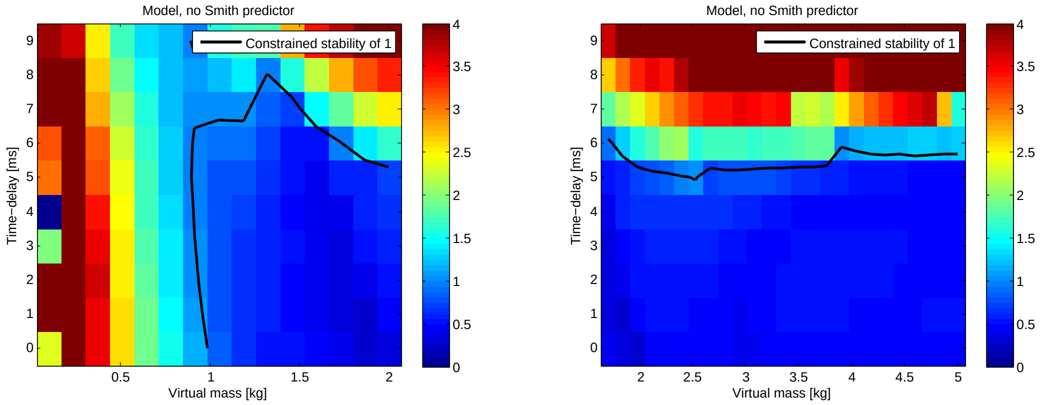
The experiments here were simple: once calibrated, the block was given an impulse of 0.025Ns to send it towards the wall where it would collide. Four seconds later, it was given another impulse but now twice as strong. Again, after four seconds it was given a third impulse, three times as strong. This would be repeated for ten different time-delays, varying from 0ms to 9ms and 25 different virtual masses, varying from 1.7kg to 5kg. Prelimanry tests showed that going lower than 1.7kg gave little extra insight, as the system was often unstable before even collide. This means that a safety precaution would stop the simulation (that being either the Control Software or the numerical model), usually due to too much current going to the actuator or too high velocities.

These experiments were done with and without a Smith predictor. Whenever a collision triggered a safety precaution which stopped the simulation it was considered void and the constrained stability η was given the value 0.

7.2.1 The Model

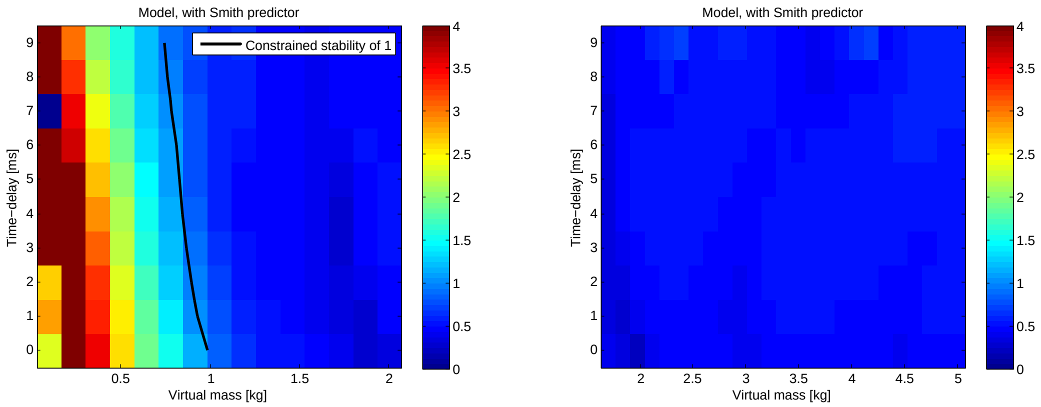
Figures 7.2b and 7.3b show the results of first experiments done with the model, with and without the Smith predictor. A constrained stability of 1 means that the robot goes out with the same speed as it goes in.

These results were considered uninteresting as the virtual masses have little effect on the stability. Figures 7.2a and 7.3a show the results of more experiments done with different virtual masses, namely virtual masses of 0.5kg to 1.75kg.



(a) (b)

Figure 7.2: Constrained stability (the colours depict the values) of the Admittance Control of the model without the Smith predictor



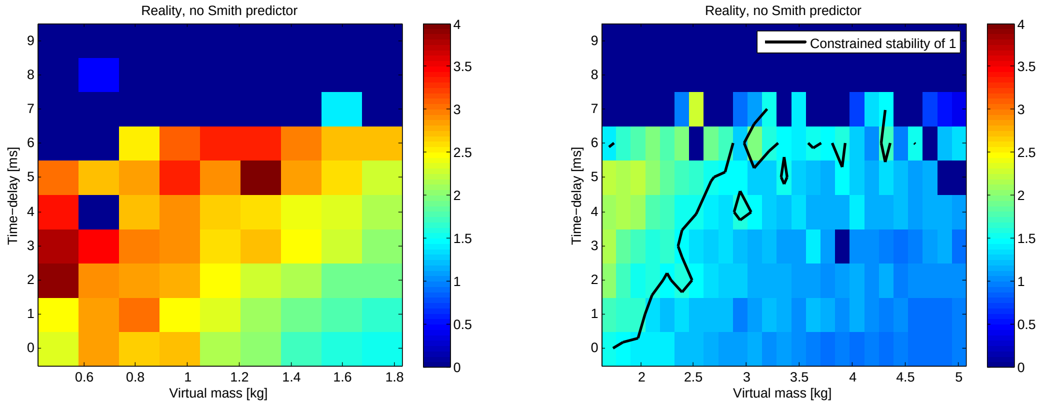
(a) (b)

Figure 7.3: Constrained stability (the colours depict the values) of the Admittance Control of the model with the Smith predictor

7.2.2 The Set-Up

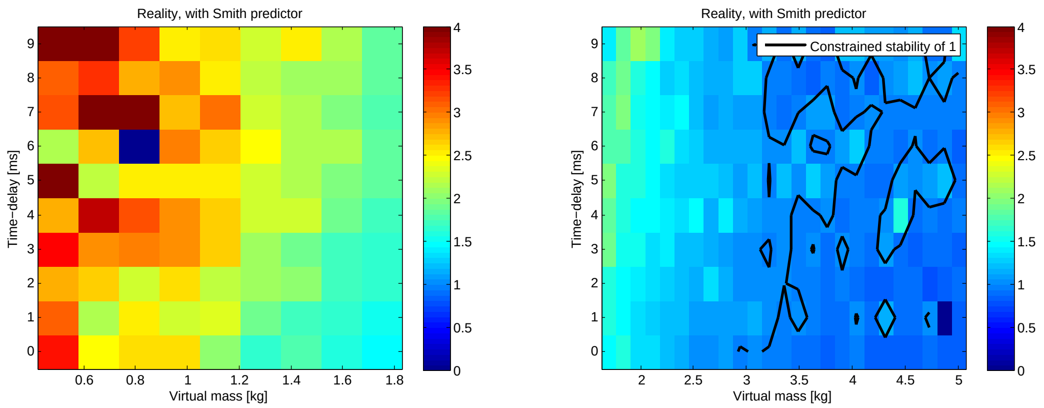
Figures 7.4b and 7.5b show the results of the first experiments done with the model, with and without the Smith predictor. The dark blue values (where $\eta = 0$) indicate that experiment was terminated prematurely due to safety precautions.

Figures 7.4a and 7.5a show the results from the second set of experiments with virtual masses of 0.5kg to 1.75kg.



(a) (b)

Figure 7.4: Constrained stability (the colours depict the values) of the Admittance Control of the set-up without the Smith



(a) (b)

Figure 7.5: Constrained stability (the colours depict the values) of the Admittance Control of the set-up with the Smith

Chapter 8

Analysis

This chapter analyses the results produced in the different experiments. First time-delays and the Smith predictor are discussed, followed by a comparison between the numerical model and the real set-up.

8.1 Time-Delay And Smith Predictor

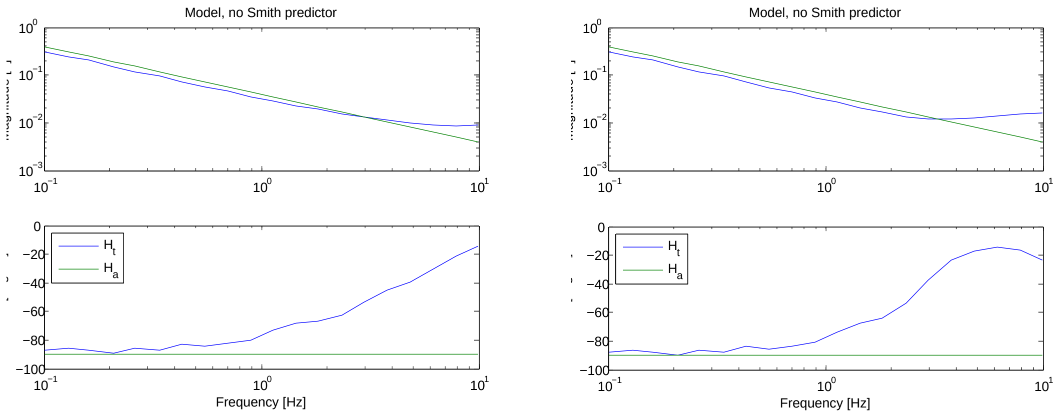
8.1.1 Haptic Quality

Without an environment to worry about, the Admittance Control can be expected to stay stable with good haptic quality, even at low virtual masses. The model shows this nicely in both figs. 7.1a and 7.1b. It is also shown that at higher virtual masses ϵ increases and the haptic quality drops. An explanation for this could be found with noise on the tachometer. At higher virtual masses, the robot will move less at given forces and move slower. The tachometer will measure lower velocity and this means that noise has a bigger impact on the results, relatively.

Time-delay was expected to cause problems and this is the case. There is notable trend that haptic quality drops with higher time-delays. For a given virtual mass of 3.5kg, a space of 10ms time-delay causes the ϵ to go from around 5 to 7.

The results becomes interesting with the addition of the Smith predictor, which works negatively rather than positively. Especially at the higher time-delays (above $\tau = 5ms$) the haptic quality effectively halves (ϵ doubles) instead of increasing. This is a surprising result at first glance.

An explanation for the lower haptic quality can be found when looking at comparable Bode plots. Take fig. 8.1 which shows the Bode plots of the two worst-case scenarios from the results: Admittance Control with $\tau = 9$ms and $M_v = 4$kg. For the most part H_t shows a near-ideal response and behaves as if the virtual mass is slightly heavier than m_v is. Without and with the Smith predictor, this is the case. Only with frequencies above 3Hz does H_t start overreacting and the difference gets bigger. This is due to an increasing phase shift (it becomes more positive) and this is worse with the Smith predictor. Apparently the extra loop that the Smith predictor adds to the Admittance Control (fig. 4.9) affects the phase of H_t.



(a) without Smith predictor (b) with Smith predictor

Figure 8.1: Bode plots of H_t and H_a of Admittance Control with $\tau = 9$ms and $M_v = 4$kg, without and with Smith predictor.

Figure 8.1 shows that the haptic quality of the Admittance Control, with or without Smith predictor, is still very good for all imposed time-delays or virtual masses at the lower frequencies. This is only the result of the model, though, as doing experiments in reality was not possible due to the excessive effect of noise. Thus little can be said about the set-up.

8.1.2 Constrained Stability

Time-delay has a negative effect on the system's constrained stability. Both the model and the set-up show that with time-delays above 6ms to 7ms the situation becomes unworkable. In the model the robot experiences great velocities and in reality it simply becomes too unsafe to continue using the robot.

When the tests were done in reality, not only collisions caused the safety precautions to go off. When the time-delays were above 7ms, it was observed that before the Control Software was able to send an impulse to the set-up, the set-up flew out of control. It showed clear instability in a free-floating situation ($H_e = 0$) and this indicates very low haptic quality. Though untested, it is safe to assume that time-delay also relevantly affects haptic quality, even at low time-delays such as 5ms.

Time-delay affects the system separately from the virtual mass, as clearly shown in fig. 7.2a. Reality does not show this as strongly, but it can be noted that low virtual masses rarely stopped the experiments from completing: no safety precautions are visible in the area with low time-delay. Instead, the moment time-delays are more than 6-7ms, safety precautions constantly kick in (figs. 7.4a and 7.4b) when there is no Smith predictor to negate the negative effects.

When the Smith predictor is added to the Control Software and numerical model, time-delay appears to suddenly have little to no effect on the constrained stability. The model, having no safety precautions to worry about, certainly shows that the Smith predictor negates the effects of time-delay on constrained stability. In reality this was have been the case in part, but it did more.

As mentioned earlier, at high time-delays the set-up was unable to finish its experiments due to safety precautions stopping the runs prematurely. One of the reasons for this was low haptic quality where the Control Software would become unstable in a free-floating environment ($H_e = 0$). When the Smith predictor got included, it was observed that this instability was greatly reduced and the experiments were now possible (the collisions could occur). In other words, the Smith predictor negated the effects of time-delay on the haptic quality as well as the negative effects on constrained stability. The former was not tested quantatively, though, and is based on observations. It also contradicts the results from the model with experiments done on haptic quality. Nonetheless, it is safe to say that the Smith predictor improved the performance on Admittance Control for both the numerical model and the set-up in the case of constrained stability.

8.2 The Model Versus Reality

The numerical model predicted very well how the real set-up would act. According to the model, virtual masses below 0.5kg cause problems with the constrained stability with ratios increasing rapidly. This is regardless of the effects of the Smith predictor. In reality, virtual masses around 1kg to 1.5kg start showing similar issues with the constrained stability. Though there is an offset regarding the virtual mass, the prediction matches reality impressively well. The clean fits in the identification phase (described in section 6.3) already hinted at a well-working model.
Similarly, at higher time-delays the system becomes unstable and the model predicts this well too. The model shows that time-delays above 7ms cause issues, and the real set-up no longer functioned properly at those time-delays. A difference is that the set-up mostly has incomplete results at those time-delays, which only indirectly show the instability. On the other hand, when the method predicted more stable results, the real results confirmed that trend. Qualitatively, the model and reality most agree. The model gives results which are in the same order of scale as the set-up.
To conclude, the model is considered useful and functional. It was gives results which are in the same order of magnitude and there are enough similarities to justify its use.

Chapter 9

Conclusions

These are the conclusions of this research:

- Time-delay, even at low values, causes instability in this Admittance Controlled robot. Constrained stability is proven to be affected and haptic quality is decreased (the latter is only shown using numerical models).
- Adding a Smith predictor to the Admittance Control greatly negates the effects of time-delay on constrained stability. Knowledge of the magnitude of the time-delay is required for a proper working Smith predictor. The Smith predictor descreases haptic quality according to numerical models.
- The (numerical) model of the set-up proved functional and useful.

Chapter 10

Discussion

Even though the numerical model properly predicted the results, it gives no insight into the individual parts of the set-up. It functions properly and experiments can use it to predict what the real set-up will do, overal. The numerical model was made using a model of the plant and therein lies one of the problems: the set-up (which was defined to be the plant) was treated as a black box in the process of identifying it. When a certain reference force went to the actuator, a certain velocity was measured. These results were used to make the model of the plant, but they ignore the internal processes of the plant. Thus, there is little knowledge about the reasons why the plant works the way it does and any potential disparities can be hard to explain as a result of that. For more precise predictions, though, more insight on the plant is needed.

There is still little known about the effect of time-delay on haptic quality. Proper testing on the set-up was not possible, but the experiments on the model and the observations from the experiments with the constrained stability suggest that haptic quality decreases with time-delay. Yet the Smith predictor appeared to worsen the situation (fig. 7.1b) and improve the situation (observations during experiments on constrained stability). Future research can focus on finding an answer to this contradiction by designing a set-up for it. Such a set-up would have to have decreased noise and more space for the robot to move. In short, a large Haptic Master$^{\mathrm{TM}}$ would be advisable.

The Smith predictor improved the situation with environment impedance more than was expected (section 4.3.2 and fig. 4.11). Even at the worst time-delays and lowest virtual masses (which was considered the worst scenario) the robot had a clean collision (meaning that there was constrained stability ratio of around 1) with the environment. This was the case for both the numerical model and the set-up. It is an easy fix and even if designed based on many assumptions about the system (such as ideal sensors or perfect force transfer to the robot) it shows good results.

Bibliography

[1] Gabriel Aguirre-Ollinger, J Edward Colgate, Michael A Peshkin, and Ambarish Goswami. Inertia compensation control of a one-degree-of-freedom exoskeleton for lower-limb assistance: Initial experiments. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 20(1):68–77, 2012.

[2] Elmo Motion Control. Violin. http://www.elmomc.com/products/vio-main.htm, 2013. [Online; accessed 1 august 2013].

[3] Merriam-Webster Dictionary. Robot. http://www.merriam-webster.com/dictionary/robot, 2013. [Online; accessed 26 juli 2013].

[4] Neville Hogan. Impedance control: An approach to manipulation. In *American Control Conference, 1984*, pages 304–313. IEEE, 1984.

[5] Inc Intuitive Surgical. da Vinci Surgery. http://www.davincisurgery.com/da-vinci-surgery/da-vinci-surgical-system/, 2013. [Online; accessed 26 juli 2013].

[6] MOOG™. Simodont®. http://www.moog.com/markets/medical-dental-simulation/haptic-technology-in-the-moog-simodont-dental-trainer/, 2013. [Online; accessed 29 juli 2013].

[7] University of Michigan. Control Tutorials for Matlab & Simulink. http://ctms.engin.umich.edu/CTMS/, 2013. [Online; accessed 10 juli 2013].

[8] Universiteit Twente. Biomechanical Engineering. http://en.wikipedia.org/wiki/Robot, 2013. [Online; accessed 29 juli 2013].

[9] Herman van der Kooij, Jan Veneman, and Ralf Ekkelenkamp. Design of a compliantly actuated exo-skeleton for an impedance controlled gait trainer robot. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pages 189–193. IEEE, 2006.

[10] RQ Van der Linde and P Lammertse. Hapticmaster–a generic force controlled robot for human interaction. *Industrial Robot: An International Journal*, 30(6):515–524, 2003.

[11] Nationaal Kompas Volksgezondheid. Hartvaatstelsel,Beroerte. http://www.nationaalkompas.nl/gezondheid-en-ziekte/ziekten-en-aandoeningen/hartvaatstelsel/beroerte/, 2013. [Online; accessed 26 juli 2013].

[12] Wikipedia. Haptic Technology. http://en.wikipedia.org/wiki/Haptic technology, 2013. [Online; accessed 28 juli 2013].

[13] Wikipedia. Haptics. http://en.wikipedia.org/wiki/Haptics, 2013. [Online; accessed 28 juli 2013].

[14] Wikipedia. Laplace transform. http://en.wikipedia.org/wiki/Laplace transform, 2013. [Online; accessed 7 august 2013].

[15] Wikipedia. Poles. http://en.wikipedia.org/wiki/Pole (complex analysis), 2013. [Online; accessed 29 juli 2013].

[16] Wikipedia. Robot. http://en.wikipedia.org/wiki/Robot, 2013. [Online; accessed 26 juli 2013].

[17] Wikipedia. Z-Transform. http://en.wikipedia.org/wiki/Z transform, 2013. [Online; accessed 7 august 2013].

[18] Wired. Engineers develop haptic feedback system for keyhole surgery. http://www.wired.co.uk/news/archive/2011-08/15/haptic-feedback-keyhole-surgery, 2013. [Online; accessed 29 juli 2013].

[19] George W Younkin. Compensating structural dynamics for servo driven industrial machines with acceleration feedback. In *Industry Applications Conference, 2004. 39th IAS Annual Meeting. Conference Record of the 2004 IEEE*, volume 3, pages 1881–1890. IEEE, 2004.