RAM ROBOTICS AND MECHATRONICS

ENERGY-TANK BASED TDPC FOR A HAPTIC SETUP

J. (Jirne) Snijders

BSC ASSIGNMENT

Committee:
dr. ir. D. Dresscher
L.B.L. Lenders, MSc
dr. ing. G. Englebienne

July, 2024

024RaM2024
Robotics and Mechatronics
EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE

UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Abstract

This research creates an energy-tank based Time Domain Passivity Controller (TDPC) for a DIY telemanipulation controller. The energy-tank-based TDPC aims to ensure passivity within the DIY telemanipulation controller by counteracting instability caused by time delays. The following research question was created: How can the energy-tank based method be integrated into the DIY telemanipulation controller kit to maintain system stability by counteracting the adverse effects of time delays? Literature research was done about already existing energy-tank based TDPCs. Based on this, an analysis was written about the implemented energy-tank based TDPC. The energy-tank based TDPC was tested on a DIY telemanipulation controller created by avatar.report. In these tests, it was found that without the energy-tank based TDPC, the system became unstable with a time delay of 8ms. When implementing the energy-tank based TDPC, the system was able to remain passive for a time delay of 8ms. Furthermore, it was found that there is still room for improvement within the implemented TDPC. This room for improvement is discussed and future recommendations are made. Ultimately, the research question is answered and a successful energy-tank based TDPC is created for a time delay of at least 8ms.

Contents

1 Introduction

1.1 Context

There is a turning point in society in which robotic avatar technology is becoming increasingly popular. Robotic avatars are a kind of robot which can be controlled from afar. This has many different applications, for example, healthcare or education. However, at the time, robotic avatar technology is not yet accessible to everyone. To change this, avatar.report, a trend-watching and knowledge authority on robotic avatars (avatars.report (n.d.)), aims to provide online pencasts explaining robotic avatars. Additionally, avatar.report connects these pencasts to an affordable DIY home kit focused on telemanipulation. However, parts of the controller still need to be created for this DIY home kit. One of these parts is the Time-Domain Passivity Control (TDPC). TDPC tackles a highly significant problem in a haptic interface design; namely, instability caused by time delays (Hannaford and Ryu (2002))

1.2 Problem Statement

A telemanipulation system consists of two main systems: a master system and a slave system, each consisting of a physical device and a corresponding controller. These systems are each located in a different place; therefore, the communication between them has a certain amount of time delay. This delay is further reinforced by factors such as signal processing or network congestion (Franken et al. (2011)). Time delays can result in a distorted perception for the user and additionally, it can cause energy generation within the system. If this energy generation is left uncontrolled, the system becomes unstable resulting in decreased transparency and safety issues (avatars.report (n.d.)). To counteract this issue, a TDPC can be used to maintain passivity in the system. Passivity refers to having no uncontrolled energy generation within the system.

1.3 Goal

For this bachelor assignment, I will design a TDPC for the DIY home kit by integrating the energy tank-based method. My research question is:

> How can the energy-tank based method be integrated into the DIY telemanipulation controller kit to maintain system stability by counteracting the adverse effects of time delays?

This main research question is defined into several sub-questions:

- How are currently existing energy-tank based TDPCs designed?
- How can an energy-tank based TDPC be developed to be user-friendly, easily understandable, well-documented, and connected to the pencasts of avatar.report?
- How can the energy-tank based method be optimized to maximize the effectiveness of counteracting the adverse effects of time delays within the DIY kit?
- How can the software be designed with the limiting factors of the DIY kits hardware?

2 Background

2.1 Time Domain Passivity Controller

2.1.1 General Architecture

As discussed in chapter 1, a telemanipulation controller consists of a master system and a slave system. For each system, the power port can be expressed in Force (F) and velocity ($\dot{x}$) with power (P) being the product of the two. Within each side of the system, the master and the slave side, three power ports can be detected:

1. The master and slave interaction port: $P_{m.i}$ and $P_{s.i}$
2. The master and slave control port: $P_{m.c}$ and $P_{s.c}$
3. The master and slave dynamics port: $P_{m.d}$ and $P_{s.d}$

A visual representation of this can be seen in fig. 2.1.

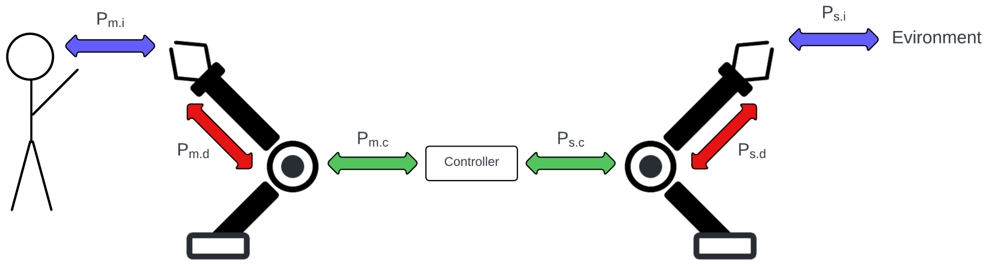

Figure 2.1: Power Ports

These three power ports have a rigid connection, which implies two things:

- Same velocities: $\dot{x}_{m.i} = \dot{x}_{m.c} = \dot{x}_{m.d}$ and $\dot{x}_{s.i} = \dot{x}_{s.c} = \dot{x}_{s.d}$
- Sum of forces: $F_{m.i} = F_{m.c} + F_{m.d}$ and $F_{s.i} + F_{s.d} = F_{s.c}$

For the system to achieve perfect transparency, it must meet the following conditions:

- $\dot{x}_{m.i} = \dot{x}_{s.i}$
- $F_{m.i} = F_{s.i}$

Within the relation of these power ports, certain issues can impact the system's transparency, such as device dynamics and time delays.

Device Dynamics

With a good energetic connection, where the velocity and force of the slave match those of the master, the system meets the following conditions:

- $\dot{x}_{m.c} = \dot{x}_{s.c}$
- $F_{m.c} = F_{s.c}$

Combining the relations of the rigid connection between the three power ports and the relation of a good energetic connection, we can see the following:

- $\dot{x}_{m.i} = \dot{x}_{s.i}$
- $F_{m.i} = F_{s.i} + F_{s.d} + F_{m.d}$

Looking back at the condition for perfect transparency, it can be seen that the forces are not perfectly equal, reducing transparency. This is a result of the device dynamics.

Time Delays

The second problem is time delays. As discussed in chapter 1, time delays can be a result of communication, signal processing, or network congestion. In a mathematical sense, let's call this time delay T.

When sending a signal from the master side to the slave side, the relation between the velocity and force will be as follows:

- $\dot{x}_{m.i}(t) = \dot{x}_{s.i}(t+T)$
- $F_{m.i}(t) = F_{s.i}(t+T)$

Looking back at the condition for perfect transparency, it can be seen that this condition is not met. Therefore, the effect of time delay reduces transparency within the system.

2.1.2 Passivity

To limit uncontrolled energy generation, a TDPC focuses on maintaining passivity (having no uncontrolled energy generation within the system (chapter 1)). A telemanipulation controller can be seen as a big system consisting of smaller sub-systems interacting with each other. The sub-systems can be organized as follows:

1. Master
2. Master Controller
3. Communication Channel
4. Slave Controller
5. Slave

A visual representation can be seen in fig. 2.2

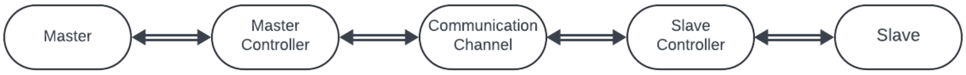

Figure 2.2: Flow Chart Sub-Systems

To create a passive system, it can be said that a system composed of passive sub-systems is passive (avatars.report (n.d.)). Looking at these subsystems, it can be assumed that most systems are already passive or can be designed such that they are passive. The issue lies in the communication channel which is generally not passive.

Within each sub-system, there are three forms of power:

1. Power storage: P_S
2. Power dissipation: P_D
3. Power generation: P_G

With the total power being the sum of all three. For a passive system, the power dissipation needs to be greater than the power generation. If this condition is met, it results in no uncontrolled energy generation and therefore a stable system (avatars.report (n.d.)). Subsequently, the relation can be written as follows:

$$\sum P = \sum P_S + \sum P_D + \sum P_G = \sum P_S + \sum P'_D \ \forall \ \sum P_D \geq -\sum P_G \tag{2.1}$$

With $\sum P'_D$ being the net dissipation. Assuming the system power generation and power dissipation are constant, we can see the following relation:

$$P_{Dmin} > P_G \Rightarrow D > \frac{P_G}{\dot{x}^2_{Dmin}} \tag{2.2}$$

By adding a dissipative element with value D the system can be made passive by ensuring that the power dissipation is always greater than the power generation. However, in a telemanipulation controller, the power generation and power dissipation is not constant. Adding a dissipative element would still work if you were to calculate it with the maximum power generated within the system. However, due to the power generation change, it would result in power dissipation when this is not necessary resulting in a decrease of transparency. Additionally, adding a dissipative element in the system adds to the device dynamics which also reduces transparency (avatars.report (n.d.)). To work around this issue, an adaptive element needs to be introduced within the system such that there is always more dissipation than generation but only a sufficient amount at the right time.

2.1.3 Energy-Tank based TDPC

One way of creating such a system is via the energy-tank based TDPC. Looking at the energy balance of a telemanipulation controller, the following can be stated about the total energy of the system at time t (H_T) to ensure passivity, assuming the initial energy of the system is zero (Franken et al. (2011)):

$$H_T(t) = H_M(t) + H_C(t) + H_S(t) \geq 0 \tag{2.3}$$

With $H_M(t)$, $H_C(t)$, and $H_S(t)$ being the energy within the master, communication, and slave systems. This states that the total energy within the system must be non-negative at all times. Additionally, to ensure passivity, the rate of change in total energy ($\dot{H}_T(t)$) at time t must always be smaller or equal to the power flowing in the controllers Franken et al. (2011):

$$\dot{H}_T(t) \leq P_M(t) + P_S(t) \tag{2.4}$$

With $P_M(t)$ and $P_S(t)$ being power flowing into the master controller and the slave controller. The conditions in eq. (2.3) and eq. (2.4) ensure passivity within the system.

A TDPC consists of a passivity observer (PO) and a passivity controller (PC). The PO monitors the flow of energy in and out of the subsystem. If the conditions for passivity are not met, it sends a signal to the PC. In response, the PC can adjust control variables within the system to ensure the passivity within the system. A visual representation of this can be seen in fig. 2.3 (avatars.report (n.d.)).

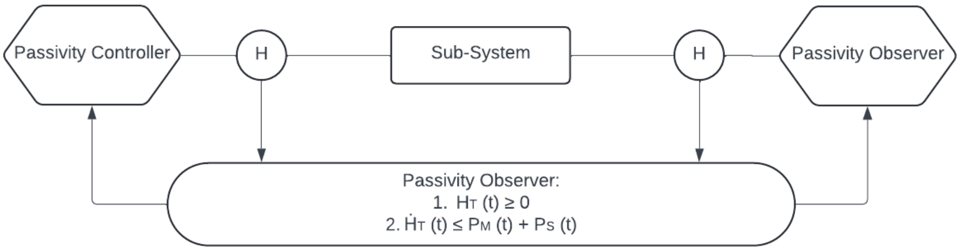

Figure 2.3: Block diagram of a TDPC (avatars.report (n.d.))

Passivity Observer: Energy-Tank

Considering the research paper of Franken et al. (2011), in both the master and the slave system, three types of energy flows can be seen:

1. Energy exchange between the system and the physical world - $\Delta H_I(k)$
2. Energy incoming from the other system - $H^+(k)$
3. Energy outgoing to the other system - $H^-(k)$

Given the controller works in discrete time, these energy flows are also in discrete time with sampling period $\overline{k}$. Additionally, the position measured is noted in discrete time as $x(k)$.

In an energy-tank based TDPC, there exist two energy tanks. One energy-tank in the master controller and on in the slave controller. The goal of these energy tanks is to maintain passivity by monitoring the available energy within the system and regulating the output such that the passivity conditions are met. Monitoring the available energy within the system is done via these three energy flows at both the master and slave sides.

The first energy flow, the energy exchange between the system and the physical world, can be calculated with the difference in position at sample time k ($\Delta x(k)$) and the torque ($\tau(\overline{k})$) of the motor at the master or slave side by the following equation:

$$\Delta H_I(k) = \tau(\overline{k})\Delta x(k) \tag{2.5}$$

Furthermore, the second energy flow is the incoming energy from the other system. This energy is sent using energy packets. An energy packet is a discrete unit of energy sent from one energy tank to the other. The incoming energy from the other system can therefore be calculated as the sum of all energy packets sent at time sample k:

$$H^+(k) = \sum_i \overline{H}(i) \tag{2.6}$$

And lastly, the outgoing energy to the other system. The equation used to determine how much energy is sent to the other system will be further discussed in section 2.1.3. For now, to maintain a passive system and a non-negative energy within the tanks, $H^-(k)$ is limited to $H(k)$. The energy within the tank at time sample $k+1$ then becomes:

$$H(k+1) = H(k) + H^+(k) - \Delta H_I(k) - H^-(k) \tag{2.7}$$

It is important for the energy tanks to not enter a state in which all energy within the tanks is depleted. To counteract this, a tank level controller (TLC) is implemented. The goal of the

TLC is to absorb extra energy within the system by increasing the desired torque output with an additional factor τ_{TLC} at the masters' side. This extra torque is introduced when the energy within the tank falls below a certain energy threshold (ϵ). The factor is calculated with the formula:

$$\tau_{TLC}(k) = -d(k)\dot{x}_M \tag{2.8}$$

With factor d(k) being determined by:

$$d(k) = \begin{cases} \alpha(\epsilon - H_M(\overline{k+1})), & \text{if } H_M(\overline{k+1}) < \epsilon \\ 0, & \text{else} \end{cases} \tag{2.9}$$

In this, α determines the intensity at which the tanks replenish energy and should always be bigger than 0 (Franken et al. (2011)).

Passivity Observer: Tank Synchronization

To further ensure passivity and improve the transparency of the telemanipulation controller, the energy tanks should be synchronised. There are several ways of synchronising the tanks but for the sake of simplicity within the system, the simple energy transfer protocol (SETP) is discussed. The SETP makes use of a variable β that transmits a fixed amount of energy from one tank to the other tank each time sample. However, updating the energy value within the tank should be split up to do so effectively. First, the energy exchange and incoming energy should be subtracted and added to the tank:

$$H(k) = H(\overline{k}) + H^+(k) - \Delta H_I(k) \tag{2.10}$$

With $H(\overline{k})$ being the previous value of $H(k)$. After this, based on the new value of $H(k)$, the outgoing energy should be determined with the following equation:

$$H^-(k) = \beta H(k) \tag{2.11}$$

Finally, the energy for the next time sample, $H(k+1)$, can be determined by:

$$H(k+1) = H(k) - H^-(k) \tag{2.12}$$

If the value of β is chosen such that $0 < \beta < 1$ and if there is no energy exchange with the physical world, the energy tanks will inevitably synchronize to the same level.

Passivity Controller

As for the passivity controller, the output can be regulated using three maximum outputs of the systems. The first maximum output must be given when there is energy available inside the tank. Therefore, the first output is calculated by:

$$\tau_{max1}(k) = \begin{cases} 0, & \text{if} H(\overline{k+1}) \leq 0 \\ |\tau_D|, & \text{Else} \end{cases} \tag{2.13}$$

With τ_D being the desired torque to match the other systems' torque. As stated in the research of Franken et al. (2011), the second maximum output can be derived from an estimation of the energy. The initial energy estimation can be done via the following equations:

$$\Delta H_I(k+1) = \tau(k+1)\dot{x}(\bar{k})\Delta T_S \tag{2.14}$$

With ΔT_S being the sampling period. Rewriting this equation, you can calculate τ_{max2} by:

$$\tau_{max2}(k) = \frac{H(k+1)}{\dot{x}(\bar{k})\Delta T_S} \tag{2.15}$$

As for the last maximum, $\tau_{max3}(k)$, this can be set to the maximum safe output. The overall maximum output $\tau_{max}(k)$ is then calculated by taking the minimum value of these three.

Furthermore, to determine the polarity of the output signal the following equation is used:

$$\tau_B = sgn(\tau_D) * |\tau_{max}(k)| \tag{2.16}$$

The final output at time sample $k+1$ of the master and slave system ends up as:

$$\begin{aligned} \tau_M(k+1) &= \tau_{B,M}(k) + \tau_{TLC}(k) \\ \tau_S(k+1) &= \tau_{B,M}(k) \end{aligned} \tag{2.17}$$

It is important to note that the theory described above of an energy-tank based TDPC is credited to Franken et al. (2011) and Hannaford and Ryu (2002).

3 Analysis

In chapter 1, the main research question is divided into several sub-questions were stated. In the analysis, I will try to answer these sub-questions.

3.1 How are currently existing energy-tank based TDPCs designed?

Existing energy-tank based TDPC, as discussed in section 2.1.3, are designed to maintain passivity within the system by managing the energy flow between the master and the slave system of a telemanipulation controller. As seen in the research papers of Franken et al. (2011) and Hannaford and Ryu (2002), the system has a master controller, slave controller, and a communication channel between them. Both the controllers consist of a passivity observer and a passivity controller.

In short, the passivity observer updates the energy tank within each discrete time step k using eqs. (2.10) and (2.12). The energy factors in this equation are calculated using eqs. (2.5), (2.6) and (2.11), with the tank synchronisation done with the SETP method. The passivity controller regulates the output of the system and the passivity controller takes the minimum value of three maximums as output. These first two maximums are calculated using eqs. (2.13) and (2.15) ($\tau_{max1} and \tau_{max2}$) and the final maximum is set to be the maximum allowed output (τ_{max3}) with the final output being eq. (2.17). It is proposed to use the same method of TDPC for the DIY kit.

A flow diagram illustrating the structure that needs to be added to the existing architecture can be found in fig. 3.1.

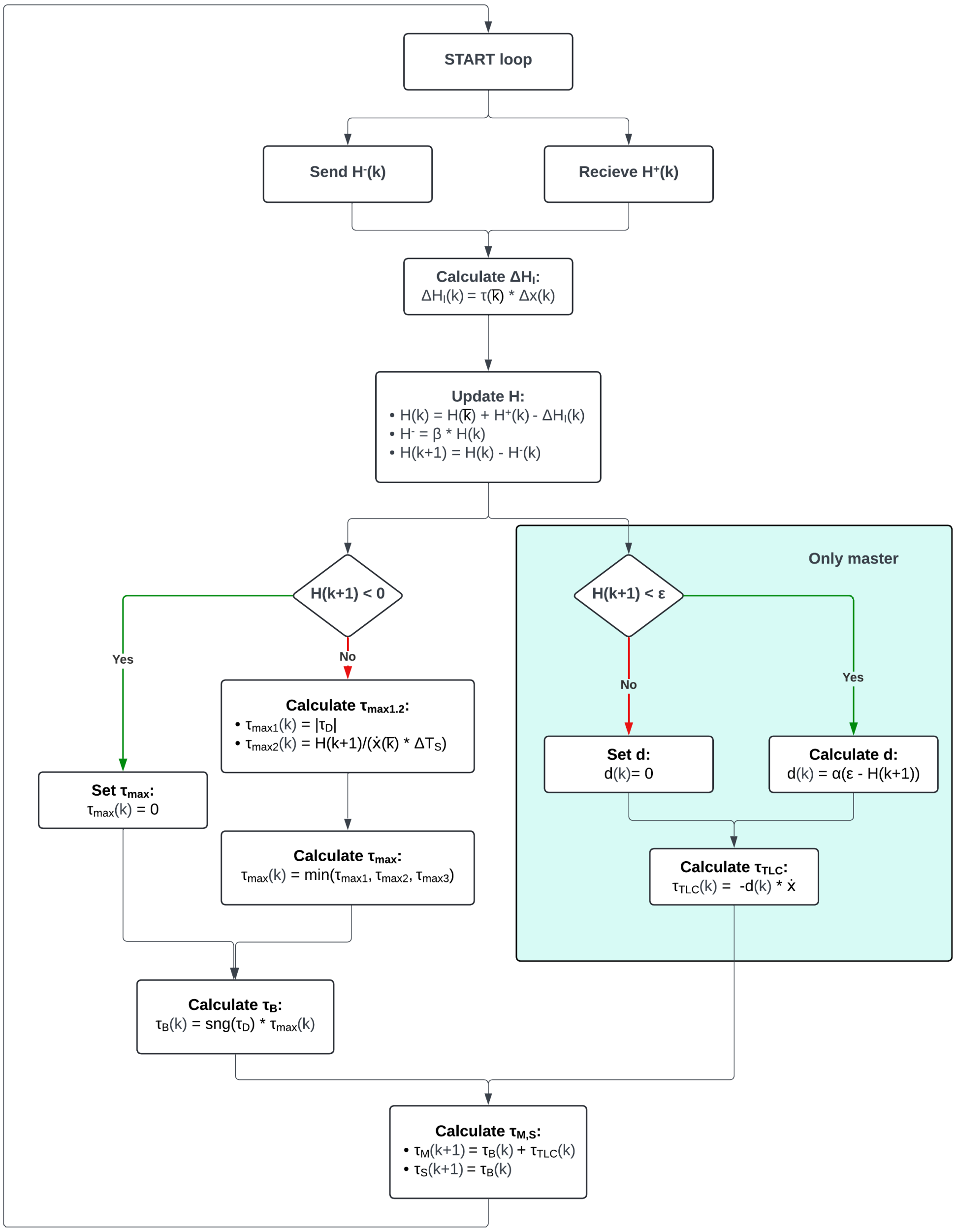
START loop
Send $H^-(k)$
Recieve $H^+(k)$
Calculate ΔH_I:
$\Delta H_I(k) = \tau(\bar{k}) * \Delta x(k)$
Update H:
• $H(k) = H(\bar{k}) + H^+(k) - \Delta H_I(k)$
• $H^- = \beta * H(k)$
• $H(k+1) = H(k) - H^-(k)$
$H(k+1) < 0$
Yes
No
Set τ_{max}:
$\tau_{max}(k) = 0$
Calculate $\tau_{max1,2}$:
• $\tau_{max1}(k) = |\tau_D|$
• $\tau_{max2}(k) = H(k+1)/(\dot{x}(\bar{k}) * \Delta T_S)$
Calculate τ_{max}:
$\tau_{max}(k) = \min(\tau_{max1}, \tau_{max2}, \tau_{max3})$
Calculate τ_B:
$\tau_B(k) = sng(\tau_D) * \tau_{max}(k)$
Only master
$H(k+1) < \varepsilon$
No
Yes
Set d:
$d(k)= 0$
Calculate d:
$d(k) = \alpha(\varepsilon - H(k+1))$
Calculate τ_{TLC}:
$\tau_{TLC}(k) = -d(k) * \dot{x}$
Calculate $\tau_{M,S}$:
• $\tau_M(k+1) = \tau_B(k) + \tau_{TLC}(k)$
• $\tau_S(k+1) = \tau_B(k)$

Figure 3.1: Block diagram of the proposed TDPC based on Franken et al. (2011); Hannaford and Ryu (2002)

3.2 How can an energy-tank based TDPC be developed to be user-friendly, easily understandable, well-documented, and connected to the pencasts of avatar.report?

As stated in section 1.1, the DIY kit from avatar.report is created to educate people on robotic avatars. Because of this, there is a high importance on developing a user-friendly, easily understandable, and well-documented software for the energy-tank based TDPC.

As will be further discussed in section 3.3, there are several variables in the software which can be tuned to optimize the effectiveness of the DIY kit. It would be educational for the used to be able to change these values easily and 'play' with these variables to see how these changes influence the stability of the system. In an earlier bachelor thesis on the DIY kit of avatar.report written by van den Boom (2023), van den Boom discussed a user interface. The proposed user interface uses sliders to help the user of the DIY learn about what these variables do. I propose that additional sliders are made in the user interface for the following variables:

- β: Being able to change β can help the user gain a better understanding of tank synchronisation speed.
- α: This variable influences the speed at which the system absorbs energy. Changing this value will create a better understanding of the tank level controller
- ϵ: Changing ϵ influences the threshold at which the TLC starts activating. Changing this value creates a better understanding of the tank level controller.

Additionally, the energy-tank levels could be visualised on the user interface as well. The most effective method and understandable would be to visualize the energy tanks as two water tanks. This way the user can see that if the system is made unstable, the energy within the tank increases and that when the system is stable, the energy decreases. Another way of visualising this would be with a spring.

A last additional slider could be regarding time delays. It would be educational to see what how different magnitudes of time delay affect the system. I would propose adding an option to the user interface to add a time delay to the system with and without the energy-tank based TDPC.

Furthermore, it is not necessary for the user to directly interact with some parts of the software. Think for example reading sensor data or writing data in the I2C bus. Therefore, these functions will be kept in the background. This reduces the likelihood of overwhelming the user with information and creates a better learning environment.

Adding these functions to the user interface will make the TDPC easier to understand and the interaction more user-friendly. Furthermore, the code added for the energy-tank TDPC to the already existing control architecture will be documented.

3.3 How can the energy-tank based method be optimized to maximize the effectiveness of counteracting the adverse effects of time delays within the DIY kit?

In the proposed TDPC discussed in section 3.1, there are several options available for optimizing the effectiveness of the TDPC.

3.3.1 Tank synchronisation

Firstly, the tank synchronisation. As proposed, the tank synchronisation for the passivity observer will be done via the simple energy transfer protocol (SETP). In the SETP, the system uses a variable, β, which can be chosen to be between 0 and 1. A higher β value causes the energy-tanks to synchronize quicker but also decreases the stability of the synchronisation. Therefore, the value of β needs to be fine-tuned such that the system stays stable. The initial value for β will be set to 0.1 and will be optimized during tests of the code.

Besides β, another change that can be made within the tank synchronisation is a change of synchronisation protocol. Besides the SETP, the research of Franken et al. (2011) proposes a second protocol of tank synchronisation. This method is called the advanced energy transfer protocol (AETP). Instead of consistently sending energy packets between the master and slave side, the AETP lets the slave side send energy requests to the master. These requests are sent when the slave tank energy level drops below a certain threshold. This threshold can subsequently be optimized for effectiveness. As a response, the master tank will send energy to the slave tank until the request is satisfied. A benefit of the AETP is that there is no consistent energy transfer. Rather, energy packets are only sent if requested. However, the AETP does increase the time necessary for transferring energy. This is due to the need to first request energy and then send energy instead of only sending energy Franken et al. (2011). If time is available, the two methods can be directly compared to see which is more effective for the DIY kit.

3.3.2 Tank Level Controller

Secondly, the tank level controller (TLC). Calculating the TLC is done with a factor d(k). In eq. (2.8) it can be seen that d(k) is calculated with two changeable variables, ϵ and α. ϵ is the energy threshold. The energy threshold is essentially the lowest point at which the tank can be before the TLC turns on. The initial value of ϵ will be set to 0.5. α is a constant influencing the speed at which the system absorbs energy when the TLC is active and will initially be set to 50. Both these values will be optimized during the testing of the code.

3.3.3 Torque and Force

The last optimization option is regarding a change in the proposed mathematical environment. In the proposed TDPC described in section 3.1, the output and input of the system are based on torque. The DIY kit, however, measured force. While it is possible to calculate the torque based on the force, it might be more efficient and effective to change the environment such that it uses force instead of torque to calculate the output of the system. To so do, a change needs to be made in calculating τ_{max2} and in calculating such that it uses force. In the research of Franken et al. (2011), this is done by using the following equation:

$$|F_{\text{max2}}(k)| = \sqrt{2H_S(\overline{k+1})K_S} \tag{3.1}$$

With K_S being the stiffness constant of a linear spring.

3.4 How can the software be designed with the limiting factors of the DIY kits hardware?

Looking back at section 1.1, the DIY kit is developed by avatar.report to be accessible for everyone. Due to this, the hardware of the DIY kit is developed to be as cheap as possible while maintaining a reasonable quality standard. However, this does result in some limitations of the hardware which need to be taken into account while designing the energy-tank based TDPC.

3.4.1 Computational power limitations

The first limitation is the Arduino Uno boards used in the DIY kit. The control architecture described in section 3.1 requires several calculations. Additionally, data constantly needs to be transferred between the two Arduinos. This needs to be done while keeping the loop frequency at 500Hz. Due to the limited computational power of the Arduinos, the code needs to be optimized in such a way that the Arduinos are capable of handling these calculations and data transfers at 500Hz.

3.4.2 Data transfer limitations

The second limitation lies in the data transfer. The data is transferred via a wire between the two Arduinos using an I2C communication. This communication works by sending bytes from one Arduino to the other. Having a higher number of bytes communicated between the Arduinos results in more time delay. Therefore, lowering the amount of bytes sent within the I2C bus decreases the communication delay between the Arduinos and would consequently be beneficial for the passivity of the haptic controller. While designing the controller, a balance needs to be found between the maximum value of the energy packets and the amount of bytes sent between the Arduinos to minimize the communication delay.

3.4.3 Force measurement limitation

The final limitation lies in the load cell and load cell amplifier used in the DIY kit. The DIY kit consists of a 2kg load cell with a HX711 load cell amplifier. Ideally, the main loop of the Arduino code runs at a relatively high frequency of around 500Hz. However, the HX711 load cell amplifier works on a 10Hz frequency. This results in either running the main loop on only 10Hz or only measuring the force once every 50 loop cycles. Both of which are not ideal. To solve this issue, instead of using force or torque for the controller, an approximation of the force/torque can be made using a constant that relates a PWM signal of the motor to force. This constant can be calculated by measuring the force on the load cell for various PWM signals and relating the force and PWM to each other by this constant. By doing so, the need for the force sensor will be eliminated and the system will be able to run on 500Hz avatars.report (n.d.); Tronics (n.d.)

4 Realization

4.1 Tank Synchronisation

4.1.1 Simple Energy Transfer Protocol

For the SETP, energy packets need to be sent between the master and the slave device. To do so the energy packets are defined as a float variable and sent as 4 bytes via the I2C bus.

Furthermore, as discussed in chapter 3, the SETP has a variable β that needs to be chosen such that $0 < \beta < 1$. Several values of β have been tested (chapter 5) and a β of 0.05 was found to be effective.

4.2 Passivity Observer

The realized passivity observer has several changes compared to the proposed passivity observer discussed in section 3.1.

4.2.1 Force Approximation

During the designing phase of the TDPC, the limitation discussed in section 3.4 regarding the force measurements were found to heavily impact the performance of the TDPC. The controller is not reliable when being ran on either 10 Hz or with the force only being measured once every 50 loops. Therefore, a constant relating the PWM to force is used.

This constant is approximated by measuring the force on the load cell for different PWM values. These values are plotted with a fitted line as can be seen in fig. 4.1. The line is fitted for values until 4N and the slope of the fitted line provides the constant. In this case, 0.0714. This approximation is done in the report of fig. 4.1. As can be seen in fig. 4.1, the force starts being measured at a PWM of 125. Furthermore, the line is fitted between a PWM of 125 and 200 Bergsma (2024). Therefore, the force approximation is only accurate between steering values of 125 - 200 and the max motor output is set to 200. Additionally, this measurement was done with a heated system (heated by using the system for 5 minutes).

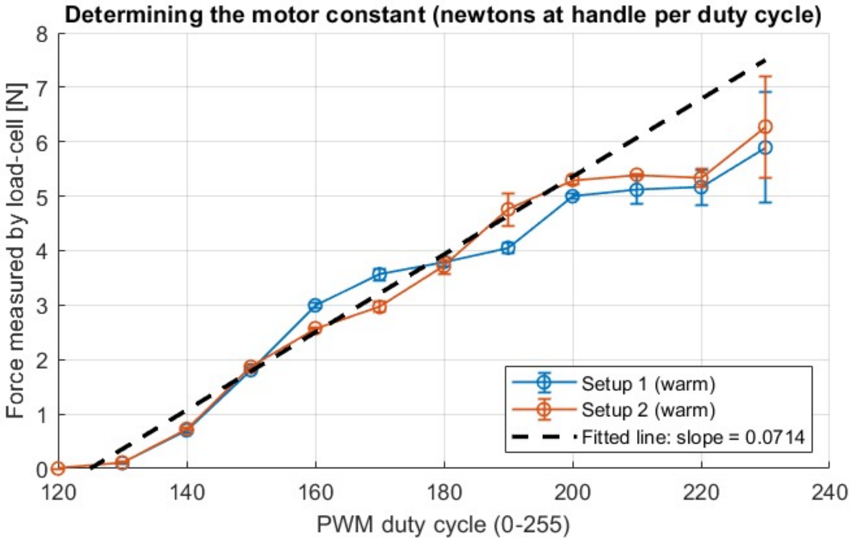

Figure 4.1: Determining the motor constant (by Bergsma (2024))

This force approximation is used in calculating ΔH_I using the following equation:

$$\Delta H_I = \Delta x * steer_{out} * for_{con} * bound \tag{4.1}$$

With:

- ΔX being the difference in position between the current and previous time step of the device
- $Steer_{out}$ being the approximated output force constrained between -200 and 200 PWM
- for_{con} being the constant 0.0714
- $bound$ being the value 0.001

Bound is a value set to constrain the energy generation and energy depletion within the system. Without the bound value, the energy tank level reaches values of around $+/-20*10^3$. I found this caused issues regarding the proposed tank level controller. The value of ϵ (the energy threshold for activating the TLC) is set to 0.05. For better passivity, it is preferred that before the energy tank reaches levels below 0, the TLC activates and starts filling the energy tank. To do so, bound was set to 0.001. This caused the energy generation and depletion within the system to become less extreme resulting in a better passivity.

4.2.2 Tank Level Controller

The tank level controller is only implemented in the master controller. Similarly to the calculation of ΔH_I, calculating the τ_{TLC} is done using the for_{con} constant. Additionally, an approximation is made for the velocity, $\dot{x}$. The equation for τ_{TLC} is as follows:

$$\tau_{TLC} = -\alpha * (\epsilon - H_M) * \frac{\Delta x}{for_{con} * \Delta T} \tag{4.2}$$

With:

- α being 0.6 (chapter 5)
- ϵ being 0.05 (chapter 5)
- $\dot{x}$ being approximated by: $\frac{\Delta x}{\Delta T}$

4.2.3 Position Difference Filter

While testing the code, I found some issues within calculating both ΔH_I and τ_{TLC}. I found that while the system remains still, both these values have sudden spikes. This caused inaccuracies within the passivity observer. In fig. 4.2 these inaccuracies can be seen in the energy tank value. These values are measured while the system remains still.

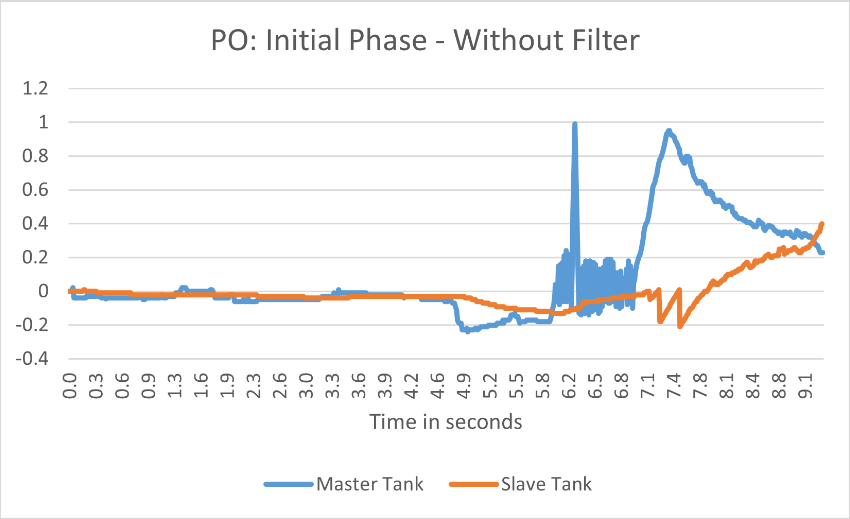

Figure 4.2: Passivity Observer, Initial Phase **Without** Filter

I found the inaccuracies to originate from the angle measurements. When calculating the position difference (the current position minus the previous position of the system), the system provides values bigger than 0 even though the system remains still. In turn, this creates sudden spikes in Δx which results in energy generation. In fig. 4.3 the absolute value of the position difference can be seen over 6.3 seconds while the system remained still.

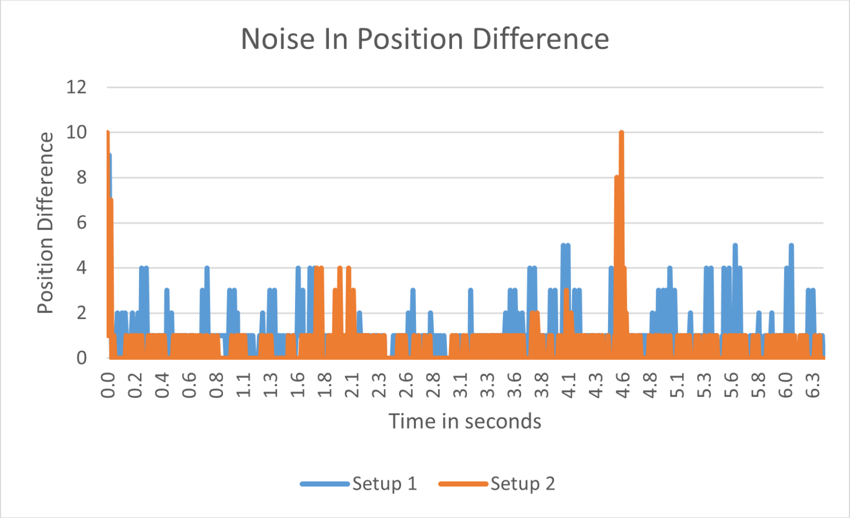

Figure 4.3: Position Difference Validation

To counteract the inaccuracies in the position difference measurements, two filters were placed in the system. Firstly, I found the initial 5 - 20 calculations of the position difference to deviate a lot, ranging to values of 100+. To counteract this an initial filter was placed in the setup phase of the code which made sure the loop only started if the position difference was calculated as 0 three times. Secondly, as can be seen in fig. 4.3, there is still noise in the system after these initial spikes. The maximum noise found was 10. Therefore, a buffer was added such that ΔH_I is only calculated when the absolute value of the position difference is above 10. This causes a decrease in accuracy for small movements but causes more reliability within the system itself.

After adding the two filters, the same measurements as in fig. 4.2 were done. The results of this measurement can be found in fig. 4.4. As can be seen, there is no more uncontrolled energy generation within the system and the energy ranks remain stable.

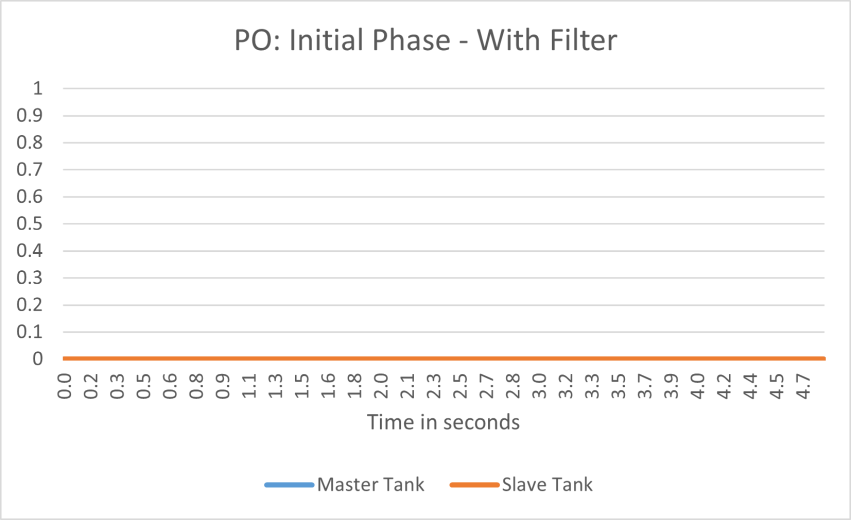

Figure 4.4: Passivity Observer, Initial Phase **With** Filter

4.3 Passivity Controller

For the passivity controller, the currently working P-cF architecture was used with some changes.

To implement the stop energy release (τ_{max1}), a part of the code was added in such a way that when the energy tank is equal to or lesser than 0 there is no output to the motor besides the TLC.

τ_{max2} has not been implemented in the passivity controller due to a lack of time.

As for τ_{max3}, this is set to be the maximum output PWM of the system, namely 200.

4.4 Hardware Setup

The telemanipulation system that the code is tested on can be seen in fig. 4.5. A more detailed explanation of the setup can be found on avatars.report (n.d.).

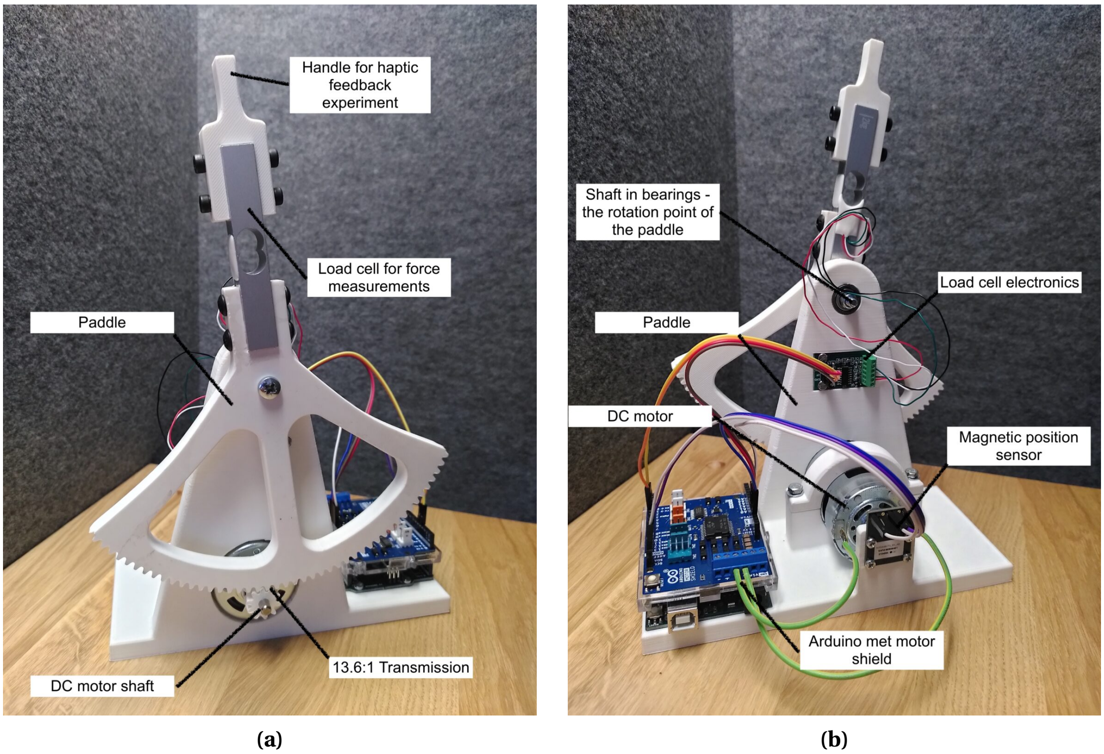

Figure 4.5: Telemanipulation system (avatars.report (n.d.))

5 Results

5.1 Simple Energy Transfer Protocol

For tank synchronisation, the SETP was used. In fig. 5.1 several different values of β are tested. The initial tank value of the master was set to 90 and the initial value of the slave was set to 10. For the test, all code besides the SETP was turned off. Based on fig. 5.1, a β of 0.05 was chosen.

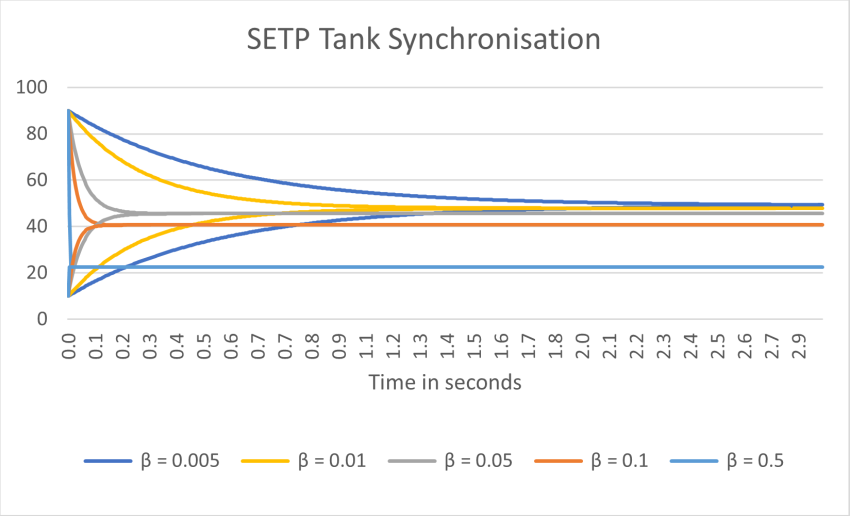

Figure 5.1: Simple Energy Transfer Protocol Results

5.2 Tank Level Controller

In fig. 5.2 both τ_{TLC} and the energy tank value of the master can be seen for the values:

- $\alpha = 0.6$
- $\epsilon = 0.05$

For this test, the master was continuously moved. As can be seen in fig. 5.2, the tank level controller provides the system with a steering value and fills up the energy tank slightly. After this, the tank level controller stops providing a steering value as expected.

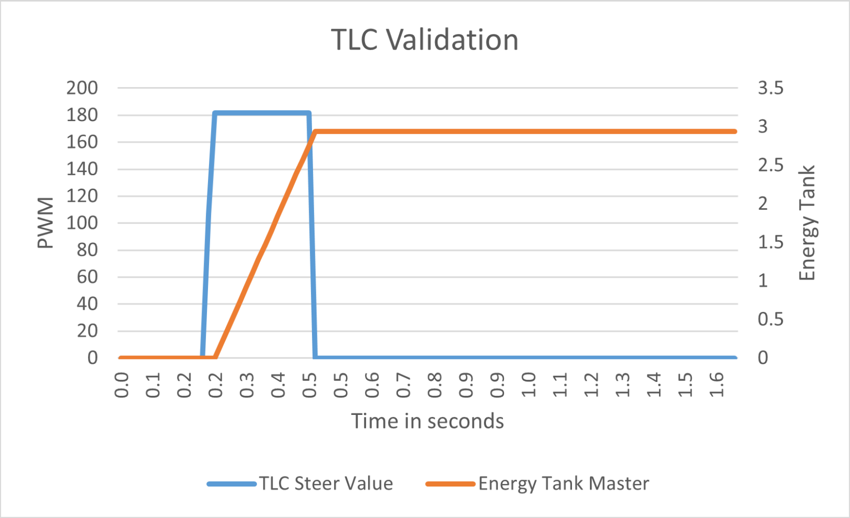

Figure 5.2: Tank Level Controller Results

5.3 Passivity Observer

In fig. 5.3 the results of the first passivity observer test can be seen. It shows the difference in energy tank value between the master and slave system can be seen. In this test, the passivity controller is disconnected and the setups are moved together by hand. Furthermore, the TLC is deactivated and both tanks are given an initial value of 5.

The initial four spikes that can be observed from about 1 to 10 seconds are the results of moving the setups to equal positions 4 times. Each peak represents moving the position of the systems once. The last peak at 11 to 15 seconds comes from moving the setups in opposite positions for about 2 seconds.

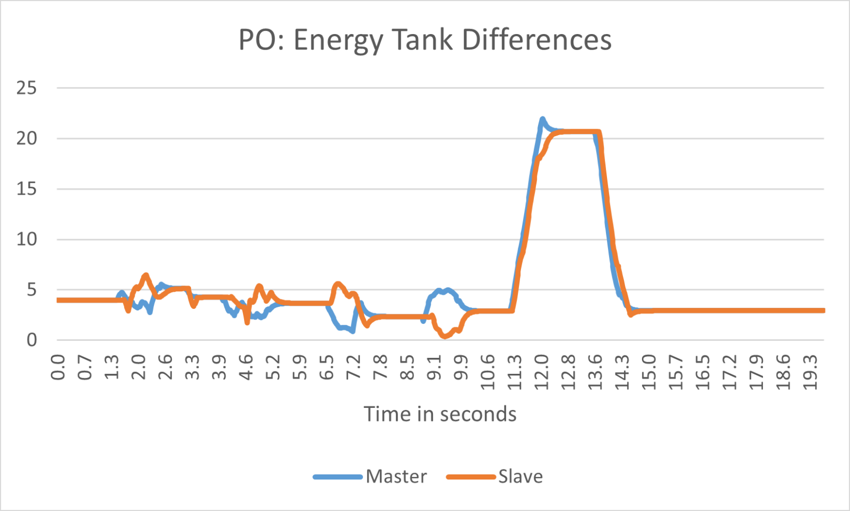

Figure 5.3: Passivity Observer Results

The results of the second test can be found in fig. 5.4. For this test, the passivity observer and the passivity controller were activated in the slave system. The energy tank of the slave system was given an initial value. The master system was deactivated entirely. The test was done by moving the position of the slave system as far away from its initial position as possible four times. For each time the position of the slave system was moved, a spike of energy can be seen in fig. 5.4.

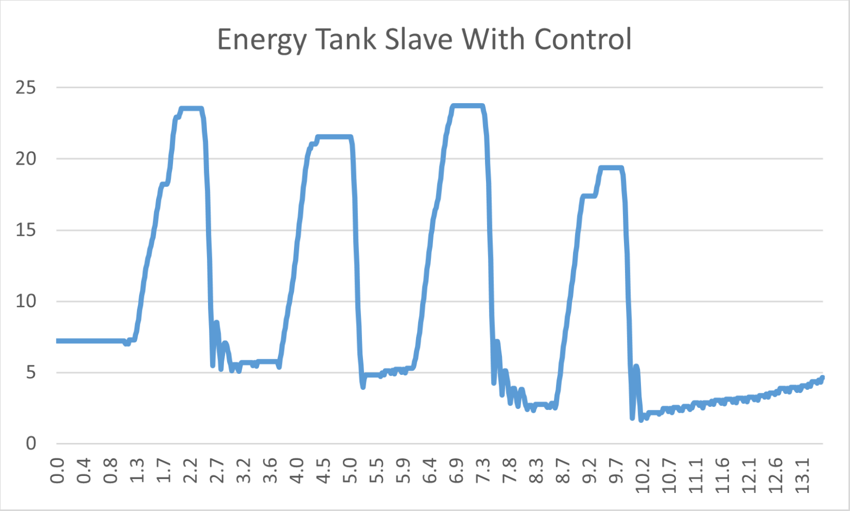

Figure 5.4: Passivity Observer Results, Slave System Energy Tank with PC

5.4 Full Architecture

5.4.1 Without Time Delay

To test the full architecture two tests were done. The results of the first test can be seen in fig. 5.5. In this test, the entire architecture was activated and the energy tanks were plotted. For the first 0 to 8 seconds, the master system was manually moved in a different position 4 times with each peak representing a different position. From 8 to 12 seconds, the master and slave system were manually forced into a different position and kept there for about 2 seconds. This resulted in a big spike of energy within the system. For the final 12 to 21 seconds, the master system was manually moved in 4 different positions again. The system behaved as expected.

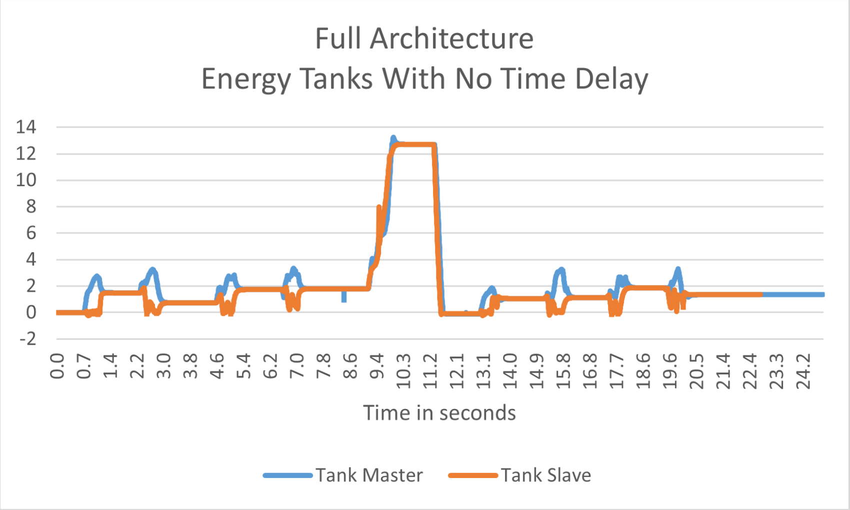

Figure 5.5: Full Architecture Energy-Tank Results **Without** Time Delay

The results of the second test can be seen in fig. 5.6. In this test, the entire architecture was once again used and the positions of the master and slave systems were plotted. In the test, the master system was manually moved for about 15 seconds. As can be seen, the slave system followed the position of the master system with each movement made. The system behaved as expected.

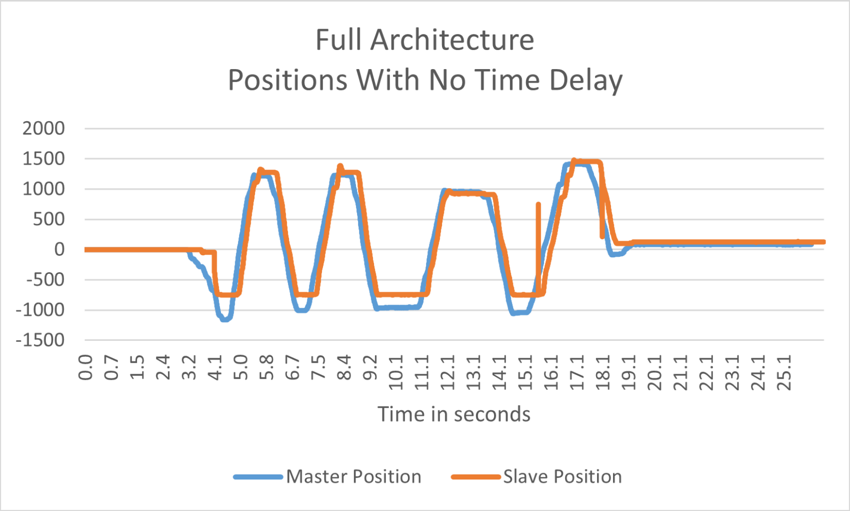

Figure 5.6: Full Architecture Position Results **Without** Time Delay

5.4.2 With Time Delay

Without Passivity Observer

Before testing the effects of time delay on the full architecture, the system was first tested without a passivity observer. The results of this test can be seen in fig. 5.7. In this test, the position of the master was manually moved between 2.5 and 3.8 seconds. After these 4 seconds, oscillating behaviour can be observed for both the master and slave system. This oscillating behaviour was exerted by the systems themselves, not by an outside force. This system was tested with an 8ms time delay.

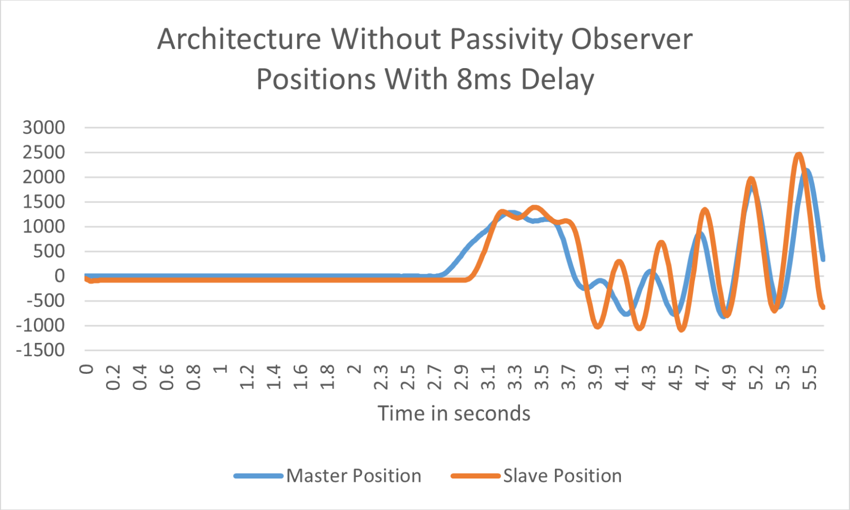

Figure 5.7: Architecture Without PO Position **With** Time Delay

With Passivity Observer

For the upcoming two test results, the full architecture was used and an 8ms time delay was forced. This includes the PO and PC. The values used in the test were:

- $\alpha = 0.6$
- $\beta = 0.05$

- $\epsilon = 0.05$
- bound = 0.001
- position difference buffer = 10
- Loop frequency = 500[Hz]
- $for_{con} = 0.0714$

Similarly to section 5.4.1, two tests were done. The results of the first test can be found in fig. 5.8. The first 3 peaks between 0 to 6 seconds are from manually moving the position of the master system 3 times. The high peak between 6 to 9 seconds comes from manually moving the master and slave system in opposite positions to force an unstable system. The final three between 9 to 14 seconds came from manually moving the master position 3 times again.

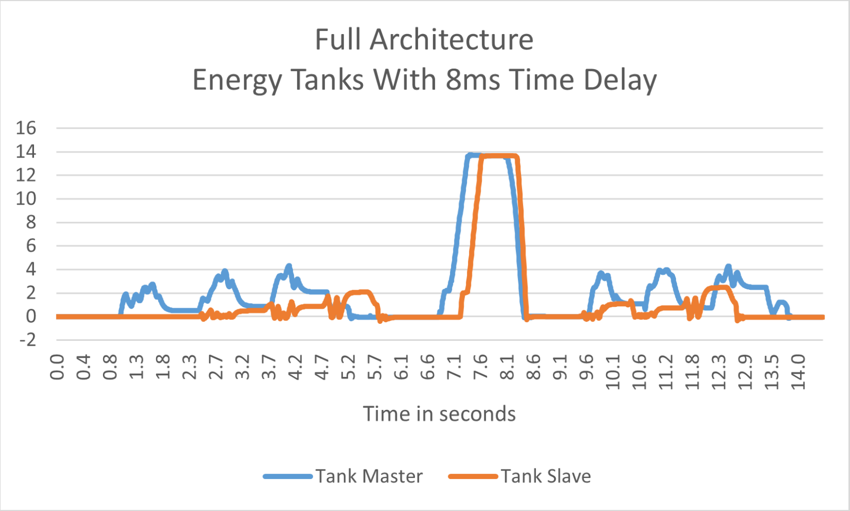

Figure 5.8: Full Architecture Energy-Tank Results **With** Time Delay

The results of the second test can be found in fig. 5.9. In this test, the master system was manually moved for about 13 seconds and both the master and slave positions were measured. As can be seen, the slave system followed the position of the master system with each movement made.

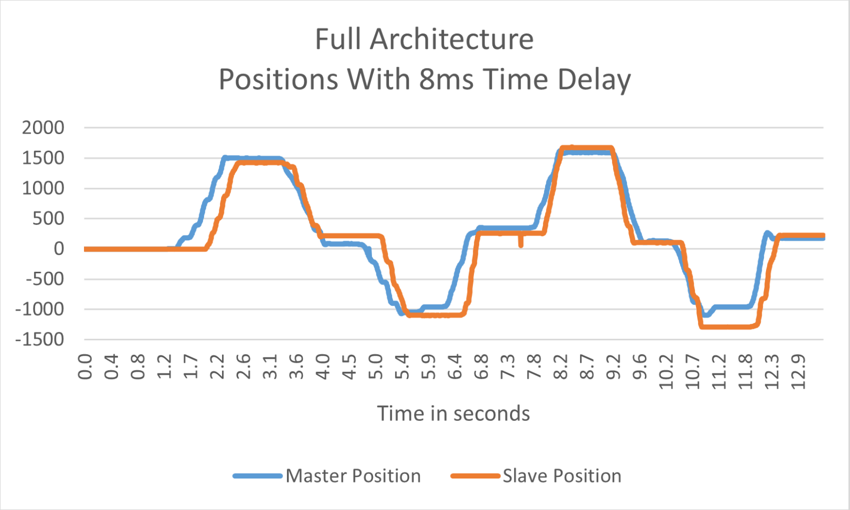

Figure 5.9: Full Architecture Position Results **With** Time Delay

6 Discussion

6.1 Simple Energy Transfer Protocol

In the test for the SETP seen in fig. 5.1, unexpected behaviour can be observed. The initial condition of the master tank was 90 and the initial condition of the slave tank was 10. Therefore, it would be expected that the final value of all measurements would converge to 50. However, as can be seen, this is not the case.

I expect this to be from the activation time of the two Arduinos. Essentially, if Arduino 1 starts its main loop 10ms before Arduino 2, Arduino 1 ends up sending several energy packets before Arduino 2 can receive them. This causes some energy to go lost within the test.

In the case that the initial conditions of the energy tanks are 0, this loss of energy packets would not be noticed due to the first couple of energy packets send equal to 0. Therefore, this is not seen as a problem for further testing.

Furthermore, due to a lack of time, there was no possibility of implementing and testing the Advance Energy Transfer Protocol (AETP). For a future recommendation, I recommend designing the AETP and testing the effectiveness of the AETP compared to that of the SETP.

6.2 Tank Level Controller

As can be seen in fig. 5.2 the TLC worked as desired. When the energy tank falls below 0.05, the TLC provides an additional steer value to absorb extra energy to fill the energy tank. When the tank is filled above the 0.05 point, the TLC stops providing this additional steer value.

Even though the TLC worked as expected, I encountered a mistake made within the code after testing was done. For $\dot{x}$, it was meant to be approximated by: $\frac{\Delta x}{\Delta T}$. However, within the code, it is now being approximated as $\Delta x * \Delta T$. Due to a lack of time I was not able to adjust the code and run the tests again. However, while the value is being approximated wrongly, the TLC still works effectively due to the value of α compensating for the wrong calculation.

For the future, I recommend changing the code in such a way that $\dot{x}$ is rightly approximated. As a result, the value of α will need to be adjusted as well to fit the calculation.

6.3 Passivity Observer

When testing the passivity observer the system is expected to behave with spring-like behaviour. In fig. 5.3 it can be seen that the system indeed acts with spring-like behaviour. When the system is passive and the setups are in the same position, as in the initial 4 peaks, the energy tank levels remain low. This is similar to a spring with no stretch. Furthermore, when the setups are put in opposite positions, the value of the tank shoots up. This can be compared to pulling a spring from both directions causing the potential energy within the spring to increase.

Additionally, if only the slave system is turned on, the system is expected to try to force itself fixed to its initial position. Similar to how a spring always tries to force itself back to its initial position. This behaviour is tested in fig. 5.4. In this test, each time the system was made unstable by manually moving the slave further away from its initial position, the energy tank of the system shot up. When the system is released, it forces itself back to its initial position and the energy tank depletes again to its initial value.

Notably in the second test, each time the system returns to its initial position, the energy tank level is lower than before. This means that somewhere in the system, energy is lost. This is not expected behaviour. I expect this to be a result of parasitic effects in the motor causing energy to be depleted in the energy tank. However, this behaviour does not result in a negative effect on the passivity of the system. It does negatively affect the transparency of the system due to having more energy depletion than necessary to maintain passivity.

6.4 Full Architecture

Before testing the full architecture with an added time delay, the architecture is first tested without a passivity observer and an added time delay. This can be seen in fig. 5.7, with an added time delay of 8ms. As stated in the results, at a time of 3.8 seconds the system started oscillating on its own. This went on until the arm of the system came loose from the gear. This showed unstable behaviour. Therefore, 8ms was chosen for testing passivity on the full architecture.

The results of the full architecture (fig. 5.8 & fig. 5.9) show that the system can remain passive even with an 8ms time delay. However, the system did tend to overshoot a bit sometimes causing the arm to fall off the gear. This happened more often, especially when the motor starts heating up. Adding τ_{max2} to the architecture could help solve this issue by limiting the output. For a future recommendation, I recommend adding this to the architecture and testing whether or not this solves the issues.

Furthermore, the system has not yet been thoroughly tested for higher time delays than 8ms. For the future I recommend finding the limit of the current architecture to see until what time delay the system can remain passive.

7 Conclusion

Looking back at section 1.3, the following research question was stated:

> How can the energy-tank based method be integrated into the DIY telemanipulation controller kit to maintain system stability by counteracting the adverse effects of time delays?

In the end, an energy-tank based TDPC was successfully integrated into the DIY telemanipulation controller and tested with an 8ms time delay. The TDPC was created using the proposed architecture discussed in chapter 3. The specifics of the architecture implemented can be found in chapter 4.

While the energy-tank based TDPC ended up working as desired, chapter 6 discusses parts of the TDPC that still have room for improvement. Firstly, there was no time for adding the advanced energy transfer protocol to the system. It is recommended to add this protocol and directly compare it to the simple energy transfer protocol to see which works more effectively. Secondly, a mistake was made in the calculation of the tank level controller. For the future, it is recommended to fix this mistake and adjust the value of α to create a correctly working tank level controller. Thirdly, due to a small energy loss in the system, likely to be caused by parasitic effects, there is a reduction in transparency of the system. Fourthly, when the motor of the DIY telemanipulation controller starts heating, the system tends to overshoot sometimes resulting in the arm falling off its gears. And finally, the system has not been tested for more than 8ms. For the future, I recommend testing to find the maximum time delay that the current TDPC can handle.

In the end, an energy-tank based TDPC was integrated into the DIY telemanipulation controller. The TDPC was created using the proposed architecture discussed in chapter 3. Furthermore, several tests were done with the TDPC to show its functionality. The results of these tests can be found in chapter 5 and as can be seen, the TDPC was able to successfully maintain passivity within the DIY telemanipulation controller for a time delay of 8ms.

A AI Statement

During the preparation of this work the author used ChatGPT & Grammarly in order to correct grammar and gain a better understanding of concepts. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the content of the work.

Bibliography

avatars.report (n.d.), Avatar robotics; explained.
https://avatars.report/

Bergsma, V. (2024), Control and Force Sensing for a DIY at Home Telemanipulation Setup.
https://www.ram.eemcs.utwente.nl/education/assignments/control-and-force-sensing-diy-home-telemanipulation-setup

van den Boom, F. (2023), Software for DIY at gome LAB for telemanipulation control.
https://www.zotero.org/groups/1664506/ram_thesis_reports/collections/KAQBK3V8/search/boom/titleCreatorYear/items/HYTWPQG7/item-details

Franken, M., S. Stramigioli, S. Misra, C. Secchi and A. Macchelli (2011), Bilateral Telemanipulation With Time Delays: A Two-Layer Approach Combining Passivity and Transparency, *Robotics, IEEE Transactions on*, pp. 741 – 756, doi:10.1109/TRO.2011.2142430.

Hannaford, B. and J.-H. Ryu (2002), Time-domain passivity control of haptic interfaces, *Robotics and Automation, IEEE Transactions on*, **vol. 18**, pp. 1 – 10, doi:10.1109/70.988969.

Tronics, T. (n.d.), Load Cell Amplifier- HX711.
https://www.tinytronics.nl/en/sensors/weight-pressure-force/load-cells/load-cell-amplifier-hx711