A Novel, Wave-based Control Architecture for Collaborative Haptic Virtual Environments

NILS JÖRGENSEN
A Novel, Wave-based Control Architecture for Collaborative Haptic Virtual Environments

Nils Jörgensen
Abstract

HAPTIC FEEDBACK, the introduction of sense-of-touch into virtual environments, has been shown to have many benefits in practical applications. One such area of interest, is the increased performance of tasks that require collaboration. In the literature however, a trade-off has been identified between transparency and consistency among remotely connected users.

The purpose of this thesis is to explore the realisation of a hybrid network architecture to try to gain the transparency benefits of a peer-to-peer (P2P) architecture, and the consistency and scalability benefits of a client-server (CS) architecture. Unlike the conventional CS case where the client only has a static local copy of the object, the hybrid architecture introduces a dynamics engine on the client side. There is still a central node in the topology, designated the Observer, which contains the central model and all information about the virtual environment. To maintain consistency, the Observer is connected to each client model by a consensus controller, which act as a virtual coupling.

Different distributed control strategies and choice of parameter distributions are investigated and evaluated in simulation. To ensure passivity, and thus stability, wave variable transforms are proposed as an alternative to power variables, which also removes the controller design from the analysis. A passivity-preserving, prediction-based reflection compensation algorithm is also proposed to improve the user experience during collaboration. The proposed solution is generalised for an arbitrary amount of users, for any degree of freedom, while the evaluation is limited to two users in a 1-DOF use-case.

Finally, a comparison is performed in simulation between the proposed hybrid architecture and the state-of-the-art P2P and CS architectures. The findings are evaluated in regards to stability, consistency, and transparency.
Sammanfattning

HAPTISK ÅTERKOPPLING är ett samlingsnamn för tekniker som introducerar känsel som en informationskanal för användare i virtuella miljöer, vilket har visat sig ha en fördelaktig påverkan på flertaliga praktiska applikationer. Ett område som visat sig särskilt intressant är de uppgifter som kräver kollaboration mellan användare. För sådana uppgifter har det inom forskningen identifierats en tydlig avvägning mellan haptisk transparens, och konsistens mellan fjärranslutna användare.

Syftet med detta examensarbete är att utforska realisationen av en hybrid nätverksarkitektur, med syfte att kombinera transparensfördeelarna från peer-to-peer, med skalbarheten och konsistensfördelarna från en klient/server-arkitektur. Till skillnad från den konventionella lösningen, där klienten endast har en statisk kopia av det virtuella föremålet, så introducerar den hybrida lösningen en fysikmotor även på clientsidan. Den centrala noden i topologin kvarblir, och innehåller en central modell samt all övrig information om den virtuella miljön. För att bibehålla konsistens, är varje klient kopplad till centralnoden med en s.k. konsensusregulator, som agerar som en virtuell fjäderkoppling.


Acknowledgements

This Master’s Thesis could not have been completed without a number of individuals, whom I would like to acknowledge and thank for their assistance, feedback and support.

First, I would like to express my great appreciation to the Mechatronics Department of Machine Design at KTH for their continuous support, with a very special thanks to my academic supervisor Dr. Binbin Lian for helping keep me on track. I am also particularly grateful to the examiner Prof. Martin Törngren for his interest in the project, and friendly consultation, as well as to Dr. Fredrik Asplund for his guidance and advice regarding the research methodology, and his help in shaping the initial direction of this thesis work.

I would also like to express warm gratitude to the entire research team at Ericsson Research Device Technologies, including the other thesis workers, for their involvement, helpful insight, and encouragement. Special thanks to my industrial supervisor Ioannis Karagiannis, M.Sc. for his eager commitment to the project from day one, and for seeing it through to the very end, to Dr. Lars Andersson for identifying the gap in the state-of-the-art which served as the foundation of this thesis, and to Dr. José Araújo for helping me get a running start during the initial few days, despite starting a little late.

Finally, I would like to extend my gratitude to Prof. Petter Krus and his team at the Division of Fluid and Mechatronic Systems (FluMeS) of Linköping University, for taking the time to present their work on TLM, and the fruitful discussions that followed.
Nomenclature

Abbreviations

BIBO  Bounded input/bounded output
CHVE  Collaborative haptic virtual environment
CS    Client/server
DOF   Degrees-of-freedom
HIP   Haptic interaction point
LTI   Linear time invariant
MR    Mixed reality
P2P   Peer-to-peer
PC    Passivity controller
PO    Passivity observer
QoE   Quality-of-experience
RTT   Round-trip time
SISO  Single input/single output
TLM   Transmission line modelling
VC    Virtual coupling
VE    Virtual environment
VO    Virtual object
VR    Virtual reality

Distributed systems

\( \mathcal{N} \)  Ordered set containing the indices for each client, \( \mathcal{N} = \{1, 2, \ldots, N\} \)
\( N \)    Number of clients (cardinality of \( \mathcal{N} \))
\( n \)   Indices used to denote a client, when \( n \in \mathcal{N} \), or the observer if \( n = 0 \)
\( x_n \)  Signal or parameter \( x \) at node no. \( n \)
General mathematics

\( t \)  Time (independent variable)
\( \theta(t) \)  Unit step function (dimensionless)
\( s \)  Complex frequency (independent variable)
\( G(s) \)  Dimensionless transfer function
\( Z(s) \)  Impedance (transfer function from flow to effort)
\( Y(s) \)  Admittance (transfer function from effort to flow)
\( C(s) \)  Controller transfer function (from flow to effort)

Port-based modelling

\( p \)  Effort
\( q \)  Flow
\( L \)  Inductance
\( R \)  Resistance
\( Z_c \)  Characteristic impedance for transmission line models
\( T \)  Time-delay in transmission line models
\( c \)  Wave characteristic in transmission line models
\( u \)  Incident wave
\( v \)  Reflected wave

Mechanical systems

\( F \)  Force (effort)
\( \dot{x} \)  Velocity (flow)
\( p \)  Momentum (integral of force)
\( x \)  Position
\( m \)  Mass (inductance)
\( d \)  Damping (resistance)
\( k \)  Stiffness (inverse capacitance)
## Contents

1 Introduction ................................................. 1
   1.1 Motivation ........................................... 1
       1.1.1 Computer haptics .............................. 2
       1.1.2 Collaborative haptic virtual environments .... 2
   1.2 Literature review .................................. 2
       1.2.1 Communication architectures ................ 3
       1.2.2 Delay compensation .......................... 4
       1.2.3 Consistency control .......................... 4
   1.3 Problem description ............................... 5
       1.3.1 Research questions ............................ 6
   1.4 Contribution ....................................... 6
   1.5 Disposition ....................................... 7

2 Preliminaries ........................................... 9
   2.1 Definitions ....................................... 9
       2.1.1 Bilateral teleoperation ....................... 9
       2.1.2 Haptic interfaces ............................ 10
       2.1.3 Haptic rendering ............................ 10
       2.1.4 Transparency ................................ 11
   2.2 Dynamical systems .............................. 12
       2.2.1 Power variables ............................... 12
   2.3 Passivity theory ................................ 12
       2.3.1 Motivating example .......................... 13
       2.3.2 Connection with stability ................... 14
       2.3.3 Passivity observer ........................... 14
   2.4 Transmission line modelling ....................... 15
       2.4.1 The lossless transmission line ............ 15
       2.4.2 Wave variables in teleoperation ............ 16
       2.4.3 Wave transform ................................ 16
       2.4.4 Condition for passivity ..................... 17
       2.4.5 Reverse transformation ....................... 18
       2.4.6 Extended implementations .................... 19
   2.5 Benchmarking setup ................................ 20
       2.5.1 Delimitations ................................. 20
Introduction

This Master’s Thesis will be completed within the scope of the Mechatronics Track of the Engineering Design Master’s Programme at the Department for Machine Design at KTH. The scope of the thesis project lies within the design of algorithms for the control of electromechanical haptic feedback systems over a network, with one or more humans in the loop, and the mechanics simulation required to provide the user with a realistic experience. This involves classical control theory and passivity analysis, network architectures, and the modelling of dynamical systems, including transmission line modelling, to name a few areas of interest.

1.1 Motivation

With the dawn of the personal computer, virtual environments (VEs) that allow users to interact with a simulated 3D world has become an attractive topic both in the academic community and the industry. While the average consumer may only be familiar with this in the form of video games or training-simulators for operating vehicles, the applications can extend to many other fields. The survey on VEs [1] lists a wide array of fields, including bioengineering and geology, oil and gas, automotive engineering, manufacturing, architectural mockup and CAD.

The user’s interaction with the virtual content is usually enhanced with visual and auditory information, and has previously been restricted to a flattened representation on a computer screen, but recent technological advances have led to an increased interest in immersive virtual reality (VR) settings [1]. Similarly, two emerging fields of research are augmented reality, and more recently, mixed reality (MR). In contrast to VR, where the real user’s visual field is completely replaced with the virtual environment, augmented reality superimposes virtual objects onto the real-world. The aim of MR is to create an immersive experience in which the virtual content co-exists with, and interacts with real-world objects in a believable way. Hardware for head-mounted displays (HMDs) increase the user immersion by combining stereoscopic visualisation and head tracking, which “helps create the illusion of virtual objects being part of the physical world” [1]. The MR applications are expected to become widespread in the future, especially within the Industry 4.0 framework [1], and with the emergence of the 5G Tactile Internet [2].
1.1.1 Computer haptics

According to [3], haptic feedback is the next critical step in developing multimedia, and the authors behind [4] suggest that haptics could provide new possibilities in areas such as medicine, industry, education, entertainment, and graphic arts. They define computer haptics as the discipline concerned with generating and rendering haptic stimuli to the human user, and the definition is contrasted with computer graphics, which similarly, deals with generating and rendering visual images.

Haptics comes from the Greek word “haptikos”, meaning to grasp, or touch [3]. Within technology, is used to describe the use of tactile sensations as a method of user interaction – enabling the user to touch, feel, and manipulate objects in the environment – in addition to seeing and hearing them. The authors of [4] emphasise that this provides a sense of immersion in the environment that is otherwise not feasible.

There exist several definitions of what constitutes haptic stimuli – some broader definitions include the use of vibration, or changes in air flow or pressure, which provide a “false-colour” rendering of the desired feedback signal. This is also sometime referred to as tactile feedback, and while it has been shown to have practical applications, it requires the user to interpret the feedback to understand the experience.

The use of the word haptic within the scope of this thesis will refer to force feedback, which uses actuators to directly affect the movement of the operator. The modality of this feedback is directly intuitive to an untrained user, and is meant to be an imitation of reality, directly increasing the believability and immersion of the experience.

1.1.2 Collaborative haptic virtual environments

The arrival of the internet has allowed multiple users, possibly from around the world, to collaborate within the same shared virtual environment. In [5] it is shown that haptic feedback significantly affects the task performance and the sense of togetherness in virtual environments. This is further confirmed in [6], where it is proposed that touch, in comparison to other sensory modalities, is more local and bidirectional, which is linked to closeness and intimacy, and improves the subjective experience. The same study also demonstrates that with the introduction of haptic feedback, time needed to complete a task can be reduced. The same study also demonstrates the positive impact of haptics on the task performance and the subjective sense of togetherness. This idea has matured into a field of research which will be referred to in this thesis as collaborative haptic virtual environments (CHVEs).

1.2 Literature review

The purpose of this chapter is to summarise the reviewed literature and thus specify the direction of the project work.

One of the contributions that this thesis aims to make to the field of research is a novel distribution of computation and communication.
1.2.1 Communication architectures

In the 1997 survey [7] on existing architectures, the authors present the various communication architectures that are used in haptic environments.

In 2018, a new survey on the state-of-the-art of haptic technology [2] revealed that the two communication paradigms upon which an overwhelming majority of CHVE’s are based, are still the centralised and distributed architectures.

The authors of [8] put forward that distributed strategies are typically utilised when minimising latency is a priority, while the centralised alternative is used where consistency is most important. This section aims to define what distinguishes them, and their respective strengths and weaknesses with regards to transparency, system stability, etc. Of course, some unique variants or hybrid alternatives are presented in [7], and have been proposed since then. The most interesting ones will also be given some attention.

The first category is the centralised, client-server (CS) architecture. The authors of [9] divide the CS-based architectures further into two main categories; the first denoted as CS-force architecture. In this setup, the virtual environment is rendered in a centralised server and all clients transmit the position of the respective user’s tool. The force feedback is calculated at the server and is transmitted back to the clients, which results in a closed loop that “resembles a traditional bilateral teleoperation system” [9]. This control system is susceptible to even small communication delays, with loss of stability as a potential outcome [10].

Most of the CS-based architectures in the modern literature, however, belong to the other main category, denoted the CS-state architecture. Here, each client has a local static copy of the virtual object and the force feedback is calculated locally based on the tool and the local object position. Each user also transmits the force, which the server uses to update the dynamic model.

Virtual environments utilising configurations from this category are commonly more robust in regard to stability, but they are still impaired by network conditions such as delay and jitter. If the user pushes against a virtual object in the local client copy, the object will not move immediately because the dynamics are simulated on the server and must be transmitted to the client over the communication network. The communication delay will cause the user to penetrate deeper (and the object to appear heavier) than the user anticipated, which leads to a loss of transparency.

In a distributed, or peer-to-peer (P2P) network, each user retains a local dynamic copy of the virtual object, with which they interact. Adopting a peer-to-peer (P2P) topology can mitigate the transparency loss from delays, since the haptic rendering is done solely based of each user’s local copy. The tool pose or interaction force is transmitted between all users, and each local copy is accelerated proportionally to the force resultant, the sum of the resulting forces. In the presence of delay, each local model will be effected by an individual (and probably unique) force resultant, which will cause the different states to grow increasingly inconsistent. All peers must try to bring the models together to maintain consistency, which can be realised with various means, some of which will be elaborated on below.

A variation on the distributed architecture shown in [7] is the Token-ring. It shares the consistency benefits of the CS approach and retains the responsiveness from local dynamics simulation, and enforces turn-taking by locking the VE to different users. As a result, only
one user may interact with a given VO at any given time, rendering collaboration impossible. Nonetheless, it is mentioned for its attempt at trying to combine the benefits of the two main paradigms.

In [8], a brand new distribution strategy is proposed, that tries to blend the advantages of a P2P network with the consistency benefits of a CS-network. The authors call it the roaming-server hybrid architecture and it relies on the fact that CHVEs in a conventional CS-distribution running on a local network performs produces “generally acceptable levels of communication delay” for haptic collision response”. The architecture utilises several servers, typically there would be one server on each each local network, that acts as the primary server for all clients on that network. The solution takes advantage of the times when no user is interacting with the VO to enforce consistency. While the authors note that users interacting over a slow connection still experience a less satisfying experience, it nonetheless stands out in its ambition to exploit the respective strengths of CS and P2P.

1.2.2 Delay compensation

One of the fundamental limitations of collaborative environments is the communication delay between users [6], [8]. The literature contains several methods on tweaking certain rendering algorithms or control laws to mitigate the effects of delays. This section provides an overview of some of the most interesting areas.

A common workaround to transparency loss in CS architectures is proposed in [11], where they adaptively change the coefficient(s) used to calculate the haptic feedback, thus preventing the impression of increased inertia of the VO. The approach is refined by Schuw-erk et.al. in [12] to take into account geometrical constraints, in [13] to handle multiple users and higher degrees of freedom, and recently in [9] for deformable objects, making it a quite mature method of delay compensation in CS.

An alternative delay-compensation scheme is proposed in [14], based on predicting the virtual object’s movements at the client side. The algorithm consists of two stages. First, the prediction stage, which “provides timely realistic force feedback as if there is no network delay”. Second, the convergence stage, where, by considering the smallest possible human-perceivable force, the the client-state of the local object gradually converges to the server-state. However, this also introducing noticeable inconsistency among the client-states.

1.2.3 Consistency control

The simplest way of maintaining consistency in the CS-case is to just forcefully update the position of each local copy to reflect the server state. Of course, this produces a very discontinuous motion and can cause several problems, potentially destabilising the system.

A more complete, but somewhat naïve approach to ensure consistency in a CS-scheme is shown in [15] which utilises a linear prediction algorithm and a buffering scheme. The prediction algorithm enables the client to compensate for packet loss, and the buffering scheme ensures that the haptic rendering is synchronised for all clients. This ensures that all clients hold off on haptic rendering until all clients are completely consistent. The downside here is that it forces all users to experience the same delay as the the user with the maximum delay.
Consistency is mostly a problem for P2P environments, since each peer’s local model can diverge from every other peer, and though it can be achieved by incrementally adding the previous movement from all peers onto the local version as done in [16], some more sophisticated methods will be expanded on below.

A novel P2P-based concept is presented in [17], that achieves high consistency between peers through the introduction of a “globally correct” state, in addition to “locally perceived” representations. The globally correct representation is maintained through the implementation of a global observer, that re-orders events from all peers, using a consistent global time coordinate frame, similar to the buffer scheme mentioned earlier in [15].

The synchronisation is performed as follows. In the absence of remote updates, each peer interacts with a local simulation which maintains high responsiveness. When updates arrive, the simulation is rolled back and “history is rewritten”, with the update inserted into the correct time, invalidating the local state. While the authors note that glitches might arise for large delays, they show that the consistent global state will always be recovered. The issue of stability is also not addressed formally.

A control-theory oriented solution includes a synchronisation controller as proposed in [18]. In this case, each peer employs a PD-type controller with a Smith predictor, that brings the different copies of the object together in a smooth, continuous motion. It can be treated as if each virtual object has a virtual coupling (spring-damper) to the other representations of it, and the difference in position drives the different versions together, asymptotically. In the stability analysis, the authors arrive at a threshold value, which for network delays of magnitude exceeding this value, stability cannot be guaranteed. It should also be noted that the Smith predictor proposed only deals with the case of linear dynamics, where rotation is not considered.

Variations on the virtual coupling schemes are proposed and evaluated in [19] for constant delays, and expanded on [20] for jitter and packet loss. One conclusion of the latter paper was however, that the consistency among users in the P2P-based schemes was definitively worse, in comparison to the CS architecture with local virtual coupling. The virtual coupling can still produce unstable results in the presence of delays, and two further variations based on passivity theory are shown in [21] to be effective at stabilising the system.

In [22], the authors propose a novel P2P control architecture for an internet-based CHVE with deformable objects, that ensures stability by enforcing discrete-time passivity for each component of the system. This work is the integration of several ideas from the same team, including their new non-iterative numerical integrator for the dynamics [23], and a passivity-enforcing consensus controller [24].

1.3 Problem description

The underlying work for this thesis is the proposal for a novel architecture for distributing control systems and communication over the internet, to enable a collaborative haptic virtual environment with multiple users manipulating the same virtual object simultaneously. The aim is to provide a high quality-of-experience and consistency among users, while maintaining stability in the presence of constant communication delay. The stability analysis is performed for linear systems, with arbitrary degrees of freedom.
mance of the proposed architecture is compared to two conventional, but state-of-the-art architectures in simulation. For evaluating the performance, continuous-time simulations of a linear test case with one degree-of-freedom (1-DOF) will be used. Much of the performance will be analysed from a qualitative perspective, by studying the overall system response in the time domain.

1.3.1 Research questions

The problem formulation within which this thesis was originally defined can be summed up by the following research questions.

1. In the hybrid architecture, what is the optimal distributed control in regard to stability, state consistency between instances, and haptic transparency, considering variations in the consensus controller and delay compensation strategy used in the haptic feedback controller?

2. How does the optimised distributed control of the hybrid architecture compare to the performance of the state-of-the-art CS and P2P architectures in regard to stability, state consistency between instances, and haptic transparency?

The answer to these questions will be addressed in Chapter 7. An overview of the proposed solution is presented in the following section.

1.4 Contribution

Unlike the typical CS case where the client only has a static local copy of the object, the proposed architecture will introduce a dynamics engine on the client side, similar to the P2P case. There will still be a central node in the topology, which will have a central model containing all the information about the virtual environment. The aforementioned central node will be named the Observer after [17].

To maintain consistency, the Observer is connected to each client model by a consensus controller, which will act as a virtual coupling (VC), as proposed in [20]. Each individual Observer-client connection can be seen as a bilateral teleoperation-connection, and terminology and solutions from that field will be used to formulate the problem. To ensure passivity, and thus stability, wave variable transforms will be shown to be effective as an alternative to power variables, as proposed in [21]. Finally, a comparison will be performed between the proposed hybrid architecture and the state-of-the-art P2P and CS architectures. The thesis will use different metrics to evaluate findings; first of all, requirements for guaranteed stability will be derived theoretically, under certain assumptions. The notion of transparency will be further defined in the next chapter, but will involve the responsiveness to user input, and the consistency of the virtual object’s instances among the connected users. Figure 1.1 shows a logical overview of the contribution of this thesis, and lists the topics which are explored.

Finally, it is worth mentioning that while the proposed solution is meant to be hardware-agnostic, all simulations have been run on MathWorks Simulink 9.1, and a sample of the implementation can be found in the Appendix.
Introduction

Figure 1.1: Overview of the contribution in this thesis

1.5 Disposition

The thesis is organised as follows. This chapter provides a look into the areas of collaborative virtual environments, teleoperation and haptics, and a provides the underlying motivation for the thesis. Chapter 2 provides an overview of the solutions found in the work already published in the field, elaborates on some of the key concepts, and presents a more in-depth problem description. Chapter 3 contains a detailed breakdown of the generalised proposed solution, as well as a theorem for guaranteeing stability, along with the first batch of experiments to determine how different controller configurations affect the overall performance. In Chapter 4, it is shown how a transformation to wave space allows to refine the proposed solution, and address some of the issues raised in the respective results subsection. One last important modification to the method is made in Chapter 5, which aims to mitigate the problem of wave reflections in the application. In Chapter 6 a comparison of the final solution and the conventional solutions is presented, and Chapter 7 contains the conclusion, discussion and future work sections.
Preliminaries

This chapter provides the relevant background knowledge about the area in which the thesis work was performed. This includes a brief overview of bilateral teleoperation, haptic technology, network solutions, and passivity theory, to name a few areas.

2.1 Definitions

This section provides the necessary definitions and central concepts that will be relevant in framing the work behind this thesis. This includes user force feedback in a broader sense, and some specific points of interest in the field.

2.1.1 Bilateral teleoperation

The earliest instances of introducing force feedback to an operator comes from telehaptics, which is a technology dating back to the 1940’s. The first electrical force-reflecting position servomechanism was built in 1954 [10], and was constructed within the scope of what is known as bilateral teleoperation, which enables the manipulation of a physical environment remotely. Applications include handling radioactive material, operating unmanned underwater vehicles, space robotics, telesurgery, control of mobile robots, to name a few [10]. Operational accuracy is achieved by providing the human operator with feedback mimicking the conditions at the remote location. Usually, this involves the operator (master) sending position/velocity information to the remote manipulator (slave), and receiving force feedback in return. Thus the adjective bilateral, meaning “to affect both parties”.

During the last half of the 20th century, a considerable effort was made investigating these systems further, particularly the loss of stability that can arise from communication delays in the system [10]. This has led to the maturation of several advanced control theory solutions, including but not limited to passivity-based control and wave variables. These methods will prove useful even in the virtual environment, and will be explored more in later sections.
2.1.2 Haptic interfaces

Common for both the real or virtual applications, the haptic interface is the kinaesthetic link between the human operator and the environment [25]. The purpose of the haptic interface is to measure the user’s movements, and return forces and/or torques, to convey a sense of presence to the user. Within the scope of this thesis, the kind of device used as the haptic interface is a robotic arm that the user grips the stylus at the end-joint with his hand and holds on to it while using it, see Figure 2.1a.

The embedded rotary encoders and potentiometers determine the pose of the stylus tip, relative to the workspace reference-frame, in real-time. The workspace is the subset of space which is reachable with the device tip, and is defined by the geometrical constraints and degrees-of-freedom (DOF) of the mechanical design. The origin of the frame is at a central location on the device, where the stylus can rest in a neutral mode.

![3D Systems Touch™ device, from [26]](image1)

![Spring model for force calculation](image2)

Figure 2.1: Fundamentals of haptic feedback

2.1.3 Haptic rendering

By interfacing the haptic device with a computer running a virtual environment, the pose of the tip can be transmitted and transformed to global coordinates within the virtual environment. The digital representation of the position of the haptic tool in the VE is commonly referred to as the Haptic Interaction Point (HIP). This point in the virtual space is the interface between the user and the virtual objects (VOs), and it is on this point that the user’s interactions are based. The explicit definition used in this thesis is taken from [3], and is cited in its entirety below.

Haptic rendering refers to the group of algorithms and techniques that are used to compute and generate forces and torques in response to interactions between the haptic interface avatar inside the virtual environment and the virtual objects populating the environment.

The exact implementation of how this is done can differ, and different methods will be explored in this thesis. However, the fundamental principle, widely adopted by the state-of-the-art, is the proportional force rendering.

When the user pushes the HIP into a virtual object, the respective force is calculated applying Hooke’s Law, by multiplying the penetration depth of the HIP into the virtual
object with a pre-defined stiffness $k$. This force $F_o$ is then applied in the simulation to the VO, accelerating the object away from the tip, in accordance with Newton’s second law. Most importantly, in conformance with Newton’s third law, the user also experiences a force $F_u$, equal in magnitude, and opposite in direction. A 1-DOF example of how this looks can be seen in Figure 2.1b.

Determining the penetration depth, however, is not a trivial task. In the 1-DOF case it might appear that way, since it is obvious that the direction of the force vector must be normal to the plane that is penetrated, but with higher degrees of freedom, the task becomes increasingly complicated. Multiple paths could have been taken to reach the same location inside the virtual cube, and without a history it cannot be known which path was taken. This becomes clear even when reconsidering Figure 2.1b within the 1-DOF framework – what if the user was originally pushing from the right side of the cube, but low virtual stiffness allowed them to “break through” so that the HIP is closer to the right side?

The author of [27] denotes the approach “vector field methods” and identifies several problems with this category of methods. In the same paper, the authors propose a novel algorithm in which the penetration depth is calculated as the absolute distance between the HIP and its projection, which remains tangent to the surface of the virtual object at all times. Like in [9], in this thesis the projection will be referred to as the virtual proxy.

Lastly, while the update rate of the device plays a major role in conveying a realistic sense of touch, this is not something that will be investigated in this thesis. It has been shown that at least 1 kHz is required to create a smooth illusion of haptic interaction [3], which is the update rate of the device used in this thesis, and a typical value for other state-of-the-art devices.

### 2.1.4 Transparency

The quality-of-experience (QoE) of a haptic application is a subjective metric on how well the haptic system provides an accurate experience to the user. This is commonly regarded as the transparency of the system between the environment and the operator [10]. Various definitions can be applied, depending on what aspects of the overall system is examined, and it is also important to distinguish between definitions used for teleoperation and definitions used in computer haptics.

Transparency in teleoperation commonly refers to ensuring that force control and position control is achieved with high accuracy in both the master and slave side [28], whereas transparency in computer haptics commonly refers to a more general realism and consistency of the experience. The authors of [18] identify two requirements that any CHVE must satisfy.

**Responsiveness** Each user must receive an immediate response to their haptic interactions.

**Consistency** Every user must share the same VE with identical state.

To summarise; in collaborative haptics, a transparent system performs well in handling multiple users interacting with the same object, and gives each user a believable interaction locally, while providing a sense of togetherness with the others. There are several ways of measuring this, both qualitatively and quantitively. The Delimitations section at the end of this chapter will detail the methods used to compare the haptic transparency of different configurations.
2.2 Dynamical systems

For illustrating dynamic behaviour and analysing stability and robustness, differential equations in the time-domain will be used, in conjunction with complex frequency-domain transfer functions.

The symbol “\(\rightarrow\)” will be used to denote the transformation from the time domain to the complex frequency domain, also called the Laplace domain.

\[
x(t) = e^{-at} \ x(0) \rightarrow X(s) = \frac{1}{s + a}.
\]

Consider some time domain signal \(y(t)\) and its transform, \(Y(s)\). For some arbitrary delay \(\tau \geq 0\) of this signal, the complex frequency-domain delay operator is defined as

\[
y(t - \tau) \rightarrow Y(s) \cdot e^{-s\tau}.
\]

Unless specified, all continuous signals are considered vectors of arbitrary dimension, and any frequency-domain functions will be transfer matrices of appropriate size. For convenience, the time dependency and the Laplace variable may also be dropped from the notation. Lastly, any time-domain signal and their transformed counterpart may be used interchangeably, even when describing transfer behaviour. As such, the closed loop form of a standard servo structure with reference \(r(t)\), plant \(G(s)\), controller \(C(s)\) and output \(y(t)\), can thus appear as

\[
y = (GC + I)^{-1} GC \ r
\]

in this thesis, where \(I\) is the appropriately sized identity matrix.

2.2.1 Power variables

Power variables are divided into effort and flow, whose product always gives the instantaneous power of the system. The quantities will differ depending on the energy domain, and can be force and velocity for a mechanical system, voltage and current for an electrical network, or pressure and volumetric flow rate in hydraulic systems, to name a few examples. The power-space in this application lies within the linear mechanical domain, and is thus represented by the tuple \((F, \dot{x})\), where both quantities are vectors representing the force and velocity, respectively.

2.3 Passivity theory

Passivity is a concept in control theory that deals with the energy of a general non-linear system, \(\dot{x} = f(x, u)\), where \(x\) is the generalised state vector, and \(u\) is the input vector. Like the case in Lyapunov theory, the (positive semidefinite) energy storage function \(V(x)\) will often take forms that represent the energy stored in inductive and capacitive components, in the case of physical systems.

If the overall energy of a system does not increase without non-zero input, it is said to be passive. Reversely, a system which “produces” energy, meaning , is said to be active.
A system is said to be passive if its overall energy is not increased at the absence of input. Reversely, a system is said to be active if the total energy can increase in the absence of input. 

Equivalently, it can be said that if the product of the input \( u \) and output \( y \) is equal to or larger than the rate of change in stored energy in the system, it is considered passive. This, including some further definitions are summarised in [29, Definition 5.3], listed below.

\[
u^T y \geq \dot{V}(x) \quad \text{system is passive}
\]
\[
u^T y = \dot{V}(x) \quad \text{system is lossless}
\]
\[
u^T y \geq \dot{V}(x) + \psi(x) \quad \text{system is strictly passive}
\]

which must hold for all values of \((x, u)\), and some positive definite function \( \psi \).

### 2.3.1 Motivating example

To provide some intuition for what passivity theory says about a dynamic system, a simple damped oscillator in the form of a linear spring-damper system will be examined. Consider the system in Figure 2.2.

![Figure 2.2: Simple damped oscillator](image)

The equation of motion can be written as

\[
m \ddot{x} + d \dot{x} + k x = F,
\]

where \( x \) is the position, \( F \) the external force, and \( m, d, k \) are the characteristic constants of the system representing mass, damping, and stiffness, respectively.

First, the system can be written in (linear) state-space form \( \dot{x} = f(x, u) \). The state-space representation is left up to the reader to do, but should incorporate the state vector \( x = (\xi, \dot{\xi})^T \), input \( u = F \), and the velocity chosen as the output; \( y = \dot{\xi} \). The reason for this choice is that the product of the input and output now represent the injected power into the system.

Secondly, choose the energy storage function to be equal to the energy stored in the spring and mass

\[
V(x) = \frac{1}{2} k \dot{x}^2 + \frac{1}{2} m \ddot{x}^2.
\]

Thirdly, the derivative \( \dot{V}(x) \) can be calculated using the chain rule

\[
\frac{dV}{dt} = \frac{\partial V}{\partial x} f(x, u) = F \dot{\xi} - d \ddot{x}^2.
\]

Finally, some interesting results can be shown, which gives a mechanical intuition.
1. For any non-negative damping, the rate of change in stored energy is always less than or equal to the injected power, thus proving the system to be passive.

2. For any positive damping, the difference between injected power and the rate of change in stored energy is equal to the power dissipated by the damper, and the system is strictly passive.

3. With no damping, the system turns into a harmonic oscillator, which oscillates with constant amplitude. No injected power is dissipated, and the system is lossless.

2.3.2 Connection with stability

In [29, Lemma 5.6] it is shown that for any strictly passive system, the origin is asymptotically stable, so the goal of enforcing passivity is ultimately about rendering the system inherently stable. Another very important quality of passive systems is shown in [29, Theorem 7.1], which proves that any interconnection of two passive systems is always itself, passive. Finally, and most importantly for this thesis, is [29, Theorem 7.2], which guarantees that for any two strictly passive interconnected systems, the origin of the closed-loop system is asymptotically stable (for $u = 0$).

To tie it together with linear time-invariant (LTI) systems, [29, Lemma 5.4] shows that any (strictly) positive real LTI system is also (strictly) passive. For a SISO system $G(s)$ to be positive real, it must have all poles in the left-half complex plane, and any poles on the imaginary axis must have multiplicity one. Lastly, a Nyquist plot of $G(j\omega)$ must lie in the closed right-half plane, which is only satisfied with a relative degree of zero or one. For a more rigorous definition of positive realness of transfer matrices, the reader is encouraged to examine [29, Definition 5.4].

Finally, in [30, Theorem 5.10] a special case of passivity of interconnected linear systems is shown. It is proven that it suffices that one system is strictly passive with finite $L^2$-gain, and the other just passive, to attain BIBO-stability.

2.3.3 Passivity observer

One common way of enforcing passivity, used by [21], utilises a passivity observer (PO) and passivity controller (PC), which can be placed in series or in parallel to the communication channel. The effectiveness of this method is also shown for teleoperation in [31]. The basic concept behind time-domain passivity-based control is to monitor the energy flowing to and from each node in the network, in real time, and act as a release valve. When the PO detects that the passivity condition would not be satisfied, the PC retains the system’s passivity by using adjustable dissipation elements.

One consequence of enforcing passivity is that interconnected systems become stable in the presence of arbitrary communication delays. This method of ensuring robust stability against unmodelled delays was shown to be very effective in the domain of bilateral teleoperation, and is a field that matured in the 1990’s [10].

It is important to note, however, that passivity and transparency are conflicting objectives in teleoperator system design [32], and significant research has been conducted, refining the algorithms to provide a higher quality of experience [33].
2.4 Transmission line modelling

Within simulation and numerical analysis, elements such as springs or hydraulic transmission lines can be modelled as bi-directional delay lines, containing a pure time delay and the impedance of the material. In this kind of modelling, the delay time in the virtual line is adjusted to equate the time it takes for the information to travel across the simulated medium.

Conversely, in the modelling of rigid mechanical systems, it is possible to replace the rigid joints with joints possessing some flexibility, and distribute the simulation for each joint. The delay of one integration step can be utilised in simulating the flexible element, as is shown in [34], where flexible joints with distributed parameters are used to simplify modelling of large mechanical systems, while simultaneously increasing the parallelism of the simulation.

Similarly, a distributed system communicating information across nodes with some inherent delay can be transformed to communicate the information as though it was a flexible medium, such as a hydraulic line or spring.

These different use-cases all fall under what is commonly referred to as transmission line modelling (TLM) [34], a paradigm of modelling which originates from modelling of hydraulic transmission lines in complex dynamical systems [35].

2.4.1 The lossless transmission line

This subsection is adapted from [35, Chapter 2], and provides the underlying physical intuition for wave variable communication. Quantities of effort and flow from any power domain can be used to describe the physical behaviour of the transmission line. For the sake of consistency, the fluid power quantities pressure $p$, and flow $q$ will be used here, which is consistent with the original notation.

Consider the lossless hydraulic transmission line shown in Figure 2.3. It is described by its capacitance $Z_c$ and the time delay $T$, for all signals passing through it.

\[
\begin{array}{c}
\text{Figure 2.3: Hydraulic transmission line adapted from [35]} \\
\end{array}
\]

The following list of properties are taken in their entirety from [35], and describe the set of equations that can be used to model the line.

- There is a time delay $T$ for all information propagating through the line
- Symmetry
- Flows entering the line will produce an increase in pressure in steady state
- Linear dynamics
Combining these properties yields
\[ p_1(t) = Z_c q_1(t) + Z_c q_2(t - T) + p_2(t - T), \]  
\[ p_2(t) = Z_c q_2(t) + Z_c q_1(t - T) + p_1(t - T). \]  

(2.1)

(2.2)

All the information that propagates from one end of the line to the other is collected as the respective wave characteristic, representing the waves traveling in each direction through the line. The wave characteristics are introduced as the variables
\[ c_1(t) = Z_c q_2(t - T) + p_2(t - T), \]  
\[ c_2(t) = Z_c q_1(t - T) + p_1(t - T). \]  

(2.3)

(2.4)

Using this notation, (2.1) and (2.2) can be rewritten as
\[ p_1(t) = Z_c q_1(t) + c_1(t) \]
\[ p_2(t) = Z_c q_2(t) + c_2(t) \]

(2.5)

(2.6)

### 2.4.2 Wave variables in teleoperation

In [36], Niemeyer and Slotine propose a passivity-based formalism to construct a teleoperation system which imitates physical systems and consequently obeys the law of energy conservation.

The formalism involves transforming the information sent over the communication network, from the power variables force and velocity, to wave variables representing the information travelling along a physical (and inherently passive) medium. The idea is to make the communication network to behave like a spring, similar to the third use-case of TLM mentioned above. It can be shown that the virtual spring’s stiffness is inversely proportional to the communication delay – no delay implies an infinitely stiff rod. The velocity of the end points of the spring is the flow variable, and since it does not contain any energy-generating components, is inherently passive (and stable). To achieve this, they use a variant of the wave characteristics that they call wave variables.

### 2.4.3 Wave transform

Consider again (2.5) and (2.6), but let the effort and flow represent quantities from the mechanical power space, namely force \( F \) and velocity \( \dot{x} \). Also, note how the positive direction for the transmission line is directed into the line. Changing to the mechanical domain, and respecting the idea of the same positive direction at both ends, gives the following changes in notation
\[ F_1(t) = p_1(t), \]
\[ F_2(t) = p_2(t), \]

\[ \dot{x}_1(t) = q_1(t) \]
\[ \dot{x}_2(t) = -q_2(t). \]

Finally, in the teleoperation case, it is often interesting to measure the magnitude of the waves being transmitted. For the measurements to make sense without explicit knowledge of the mechanical parameters, the wave characteristics are normalised in the following way
\[ v_1(t) = -\frac{c_1(t)}{\sqrt{2 Z_c}} \]
\[ u_2(t) = \frac{c_2(t)}{\sqrt{2 Z_c}}. \]
Preliminaries

where $u$ denotes the *incident* wave, and flows along the main direction, and $v$ is the *reflected* wave, and flows against the main direction. Using this transformation, the instantaneous power $P$ in any of the interconnected systems is given by

$$P = \dot{x}^T F = \frac{1}{2} u^T u - \frac{1}{2} v^T v,$$  (2.7)

The complete equations that describe the wave variable communications are then given by performing the appropriate substitutions in (2.5) and (2.6);

$$F_1(t) = Z_c \dot{x}_1(t) - \sqrt{2Zc} \; v_1(t), \quad \quad v_1(t) = v_2(t-T) = \frac{Z_c \dot{x}_2(t-T) - F_2(t-T)}{\sqrt{2Zc}},$$

$$F_2(t) = -Z_c \dot{x}_2(t) + \sqrt{2Zc} \; u_2(t), \quad \quad u_2(t) = u_1(t-T) = \frac{Z_c \dot{x}_1(t-T) + F_1(t-T)}{\sqrt{2Zc}}.$$  

Additionally, since the transmission line itself is lossless, the idea of preservation of energy gives some interesting results. Combining some of the above equations yields for example $u_1 = -v_1 + \sqrt{2Zc} \; \dot{x}_1$, which simplifies the communication of the transformed information.

### 2.4.4 Condition for passivity

The most significant reason for transforming to wave variables is that it assures passivity of the communication line, even for arbitrary delay. This section will provide a brief proof sketch, based on [37].

Consider the total power entering and leaving the transmission line, given by

$$P_{tot} = \dot{x}_1^T F_1 - \dot{x}_2^T F_2,$$

where the second term corresponds to any power flowing against the main direction, thus defining it negative. Substituting (2.7) gives

$$P_{tot} = \frac{1}{2} u_1^T u_1 - \frac{1}{2} v_1^T v_1 - \frac{1}{2} u_2^T u_2 + \frac{1}{2} v_2^T v_2$$  

as the expression of the instantaneous power at any time $t$. Integrating the power gives the energy storage function

$$V(t) = \int_0^t P_{tot} = \frac{1}{2} \int_0^t u_1^T u_1 + v_2^T v_2 \; d\tau - \frac{1}{2} \int_0^t u_2^T u_2 + v_1^T v_1 \; d\tau.$$  (2.8)

Assuming zero initial conditions, and substituting the time delay finally gives

$$V(t) = \frac{1}{2} \int_{t-T}^t u_1^T u_1 + v_2^T v_2 \; d\tau,$$  (2.9)

which is indeed a positive definite function.

In conclusion, the energy is stored in the system while the waves travel, making the transmission line not only passive but also lossless.
2.4.5 Reverse transformation

It is important to note that the wave transformation is bijective, meaning that it is always unique and invertible [38]. This means that no information is lost by encoding the variables in this fashion, and that the original power variables can always be decoded from the waves at any given time. The wave transform can be seen as a two-port system, where one wave variable and one power variable comes in, and the other wave and the other power variable leaves. This means that other control configurations are possible than just strict force-force communication.

In [37], a variation is proposed that implements a virtual tool on the slave-side. This means that the decoded variable at that side is velocity, not force, with a controller acting as a virtual coupling to the slave. Figure 2.4 shows how this communication looks in the form of a block diagram.

![Figure 2.4: Wave variable transform adapted from [39]](image)

Figure 2.4 shows the entire teleoperator system as presented in [37]. As models for the master and slave side dynamics, the complex frequency admittances $Y_1(s)$ and $Y_2(s)$ are used, respectively, and the slave-side virtual tool controller is described by the impedance $C_2(s)$. The blocks with wavy lines indicate the transformation to wave space, with examples of how this is achieved shown in Figure 2.4.

![Figure 2.5: Wave-based teleoperator with virtual tool, adapted from [37]](image)
2.4.6 Extended implementations

What has been described up to this point covers the basics of transmission line modelling within the domain of bilateral teleoperation.

In [38], the authors generalise the wave variable transform method for other robotics applications. While this article serves as a good introduction to the field, the authors also elaborate on some extensions that will prove useful for this thesis, either directly or for future development.

Firstly, while being a simple notion, it is still a useful one, and it involves placing a low-pass filter along the communication line, through which each wave must travel once. The wave filter will in this thesis be on the form

$$G^f(s) = \frac{\lambda}{s + \lambda}$$  \hspace{1cm} (2.10)

where the bandwidth is given by $\lambda$, and the superscript stands for “filter”. The filtering can be augmented with higher-order filters and feedback connections in the wave space to shape the waves to a larger degree, as is further shown in [38].

Next, the issue of position drift is addressed, which can occur through numerical errors or data loss in the communication. One approach to tackle this issue is by encoding the velocity along with the position, while transmitting only one wave variable. The solution is called wave integrals, and involves encoding the position and momentum into their respective integrals given by

$$U(t) = \int_0^t u(\tau) \, d\tau = \frac{Z_c x + p}{\sqrt{2}Z_c}$$  \hspace{1cm}  $$V(t) = \int_0^t v(\tau) \, d\tau = \frac{Z_c x - p}{\sqrt{2}Z_c}$$

The momentum $p$ is simply the integral of the force, and while this needs to be calculated locally to decode the position, the actual value is not important for the application itself. Transmitting the position is then just a matter of transmitting the wave integral as well as the original wave variable. To do this without increasing the required data rate, they can be merged together in one variable, denoted here as the composite wave variable.

As an example, consider the incident wave $u(t)$, of the system shown in Figure 2.4. Let $\bar{U}(t)$ be the composite wave variable transmitted over the network, and defined such that

$$\bar{U}_1(t) = U_1(t) + \frac{1}{\alpha} u_1(t)$$  \hspace{1cm}  $$\bar{U}_2(t) = \bar{U}_1(t - T).$$

By running the incoming composite wave through the stable filter given by

$$u_2(t) = \alpha (\bar{U}_2(t) - U_2(t))$$  \hspace{1cm}  $$U_2(t) = \int_0^t u_2(\tau) \, d\tau$$

the variables can be separated again, without any loss of information.

For further reading, [39] provides a survey over wave variable applications in bilateral teleoperation.
2.5 Benchmarking setup

For the sake of evaluating the performance of the proposed architecture, a simple 1 degree-of-freedom (1-DOF) use-case will be used as a benchmark.

Consider a stiff, homogenous cube resting on a non-smooth virtual surface, seen from above in Figure 2.6. The block can move freely along one direction, with the mass \( m \) and viscous damping \( d \).

\[ \begin{align*}
F & \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \frac{\partial x}{\partial y} & \end{align*} \]

Figure 2.6: Benchmark

The equation of motion can be written as

\[ m \ddot{x} + d \dot{x} = F, \]

where \( x \) is the position and \( F \) is the external force. Considering the velocity the output, the admittance transfer function of the system is given by

\[ Y(s) = \frac{1}{m \ s + d}, \quad (2.11) \]

which has one pole at \( s = -d/m \) and is stable for any positive values of \((m, d)\).

This is a common model for evaluation, and is proposed in [25] as a benchmark, and later used for analysis in [18], [11], [12] and [13]. All evaluations of performance in this thesis will use this mass-damper object as the basis for analysis and simulation.

2.5.1 Delimitations

It is worth-mentioning that the haptic rendering as shown in Figure 2.1b is not included in the mathematical analysis. There are several motivating factors for this. Firstly, it simplifies the analysis of consistency, since the application of shaped force signals implies that the studied dynamic behaviour is only dependent on the distributed control network itself, and not the individual haptic controller. Secondly, for any non-negative values for the haptic stiffness, the controller will be stable. For the study of the overall stability, it is straightforward to show that the stability of the interconnected system will be entirely dependent on the stability of the distributed control network, not the haptic controller. Lastly, the haptic controller will produce a force, back to the user, proportional to the VO’s position. For this reason, it is deemed sufficient to study the resulting kinematics of the object itself, since the force can be inferred from a qualitative examination of the motion of the VO, without knowing the actual haptic stiffness.
General client/observer architecture

The purpose of this chapter is to explore the realisation of a hybrid network architecture to gain the transparency benefits of a peer-to-peer (P2P) architecture, and the consistency and stability benefits of a client/server (CS) architecture.

3.1 System overview

The general idea of the client/observer architecture is to augment the existing CS scheme, and distribute more computational effort to each client, similar to that of a P2P setup. In the conventional CS setup, each user transmits their input to the server, which calculates the dynamics of the global object accordingly, and sends the updated state information back. In the P2P setup, each user manipulates their own local object, with the dynamics engine running at the user-side. Each user exchanges state information, and consensus controllers at each peer try to synchronise the local object with the other instances.

Finally, in the proposed observer/client setup, each user has a local copy with which they interact, and a central observer node that synchronises its central object with the other users. Each client then synchronises with the observer object, achieving state consistency across all instances.

Figure 3.1 shows a simplified functional diagram of each kind of network architecture, when only two users are connected. When considering the node topology, the hybrid architecture has the same star-shaped connections as the conventional CS-case, but the kind of information communicated over the network differs. Unlike the P2P setup, the consensus controller does not necessarily need to be present at each client, and variations of the controller placement are possible. Next section will address the topic of how to distribute the consensus control strategy.

3.2 Controller configuration

The connection between each client and the observer can be viewed as a variation of the classic bilateral teleoperation setup. Each local object can be seen as the master, and the observer object the slave. Perfect transparency is achieved when the kinematic information
Figure 3.1: A comparison of how the computation modules of a CHVE are distributed in the proposed and conventional architectures. Each node represents one self-contained instance, running on some target hardware. The arrows show what information is transmitted over the communication network.

(position and/or velocity) of the local object is reproduced by the observer object, and any external forces acting on the observer object is transmitted in its entirety to the local object, to be experienced by the user.

For all mechanical systems in this and any subsequent chapters, velocities are considered instead of positions. This is a common design choice in teleoperation, and stability and transparency are unaffected by this convention [32].

To realise a bilateral connection, the necessary information must be transmitted between the master and slave in both directions. Since the observer and all clients each have a local model that takes a force as input and produces a velocity as output, each distributed structure will require at least one controller. This will be denoted the consensus controller (CC), and will attempt to synchronize the clients’ states.

For this application, a simple PI controller will be used, and the connection can be seen as a virtual spring-damper between each client and the observer. The frictional damping $d$ acts as the proportional gain, and the spring stiffness $k$ acts as the integral gain, eliminating the steady-state inconsistency. A two-client example is shown in Figure 3.2.
For ideal transparency, the connection would be infinitely stiff, and the presence of all other client interactions would be experienced instantly. Because of time delays, this is not possible, and marks one of the fundamental limitations of delayed collaborative interaction. It remains to be determined which is the best distribution of the controller(s), and thus, which information to transmit. The different distributions will be referred to as the configurations, and will be evaluated in the next chapter. Abiding by the convention of teleoperation, the information flow will be denoted as outgoing-ingoing as seen from the client’s (or master’s) perspective. Table 3.1 shows a summary of the possible combinations of communicated data, and some tentative strengths and weaknesses.

### Table 3.1: Summary of different control configurations

<table>
<thead>
<tr>
<th>Controller</th>
<th>Communication</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observer-side</td>
<td>Velocity-force</td>
<td>Synchronises the observer with each (delayed) client. Most closely resembles the “virtual tool” structure from [32].</td>
</tr>
<tr>
<td>Client-side</td>
<td>Force-velocity</td>
<td>Synchronises each client with the (delayed) observer, which may be unique for each client at a given time.</td>
</tr>
<tr>
<td>Dual symmetric</td>
<td>Velocity-velocity</td>
<td>Transmits only kinematic information. Introduces the highest level of complexity in the system.</td>
</tr>
<tr>
<td>None</td>
<td>Force-force</td>
<td>Strictly feed-forward, will not necessarily converge. Not feasible for this application.</td>
</tr>
</tbody>
</table>

### 3.3 Parameter distributions

The next important design choice is how to distribute the physical properties of the dynamic model (2.11). Specifically, the object’s mass and the dynamic friction. In the state-of-the-art, when the simulation is distributed, common practice is to distribute the parameters
evenly between all simulations, “in order for the total mass felt by the user to be consistent” [19]. This approach may be a little naïve, as it does not take the communication delay into consideration, when designing for local responsiveness.

Each parameter affects the dynamic behaviour in different ways, the damping for instance, has a direct impact on the final steady-state or “terminal” velocity of the VO. This gives the object its feeling of resistance, and determines the force a user must apply to attain the equilibrium of the object for a given speed; \( v = F/d \). On the other hand, the object mass only affects the transient behaviour, giving the object its feeling of weight. The time constant of the object is proportional to the mass and inversely proportional to the damping, \( \tau = m/d \).

For the sake of the parameter design, the following assumptions will be made.

- Most objects that the user interacts with would likely have a time constant of a few milliseconds at most.
- The communication delay over the Internet will be sufficiently large, 50 ms to 200 ms according to [17].

Considering these two assumptions, the system is designed so that the mass of the virtual objects is distributed 100% to each client, and 0% to the observer, so that the system converges at the correct state as fast as possible.

Each client, and the observer, will contain an equal fraction of the total desired friction, since this affects the steady-state behaviour of all the clients.

### 3.4 Passivity analysis

This section puts forward an analysis of the system’s passivity, and the stability of the proposed architecture. The idea is adapted from [32], where a similar analysis and theorem are proposed for bilateral teleoperation systems. The overarching idea is to consider the interconnection of one client and the observer, and looking at the BIBO stability of the client model velocity \( \dot{x}_1 \), treating any other client’s interactions as an unmodelled, lumped perturbation term \( F^*_2 \).

The block diagram for the entire system can be seen in Figure 3.3, which is generalised to describe any of the possible configurations.

The closed-loop transfer function from force to velocity for the client-side can be split into two interconnected subsystems. The first system representing the client-side dynamics model, and the second representing the resulting impedance from the observer-side model and the consensus controller over the network. Between the interconnected systems is the generic controller architecture that enables any of the possible controller configurations. By setting the controllers \( C_1 \) and \( C_2 \), the desired configuration can be attained. For instance, setting \( C_1 = 0 \), gives the observer-side controller, with client impedance equal to \( Y_1 \). Figure 3.4 shows the resulting single-loop, interconnected system. Finally, based on this representation, the following theorem is proposed to guarantee passivity, and thus stability.

**Theorem 1.** Consider the feedback connection of the LTI systems shown in 3.4. Let \( S_1 = (G_2 C_1 + C_2 G_4) (I + Y_1 C_1)^{-1} Y_1 (G_1 C_2 + C_1 G_3) \) and \( S_2 = (I + Y_2 C_2)^{-1} Y_2 \). Under the assumption of bounded...
input signals $F_1$ and $F_2^*$, then all system signals will be bounded, given that the following conditions hold:

1. $S_1$ is strictly positive real.

2. $S_2$ is positive real.

3. The transfer matrices $G_2C_1 + C_2G_4$, $(I + Y_1C_1)^{-1}Y_1$, $G_1C_2 + C_1G_3$, and $(I + Y_2C_2)^{-1}Y_2$ are BIBO-stable.

Proof. The first two conditions 1 and 2 imply that the feedback loop satisfies the passivity theorem, as provided in Section 2.3.2, hence the output map from any given input is BIBO stable. Condition 3 ensures that no forbidden zero-cancellations are performed, which guarantees internal stability, such that all signals remain bounded. □
3.5 Effects of communication delay

This section also draws from [32], which demonstrates a similar conclusion. Even if the individual distributed control systems are designed to ensure stability independent of each other, it holds that if the signals transmitted between the client and the observer are delayed, condition 1 of Theorem 1 will not hold. This will be shown for the constant delay case, where the transfer functions modelling the communication, \( G_1 \) through \( G_4 \), contain the delay operator \( e^{-sT} \), where \( T \) is the one-way delay in transmitting information between the client and the observer.

The result is that the system \( S_1 \) contains a factor of \( e^{-2sT} \), and cannot be strictly positive real. This can be seen since the Nyquist curve of a delayed system will not remain entirely bound in the positive imaginary half-plane in the complex frequency domain. The conclusion is therefore, that adding communication delay to an otherwise passive design violates passivity, and in turn, nullifies the guarantee of stability.

3.5.1 Benchmark example

To give a bit more clarity, the passivity argument is applied to the 1-DOF benchmark case. Consider the case with the observer-side controller and with pure delay, such that \( C_1 = 0 \), \( C_2 = d_c + \frac{k_c}{2} \) and \( G_1 = G_2 = e^{-sT} \). All the mass is with the client, and the damping is split evenly between the two simulations. Inserting (2.11) into the theorem from Chapter 3 yields

\[
S_1(s) = \frac{1}{ms + d/2} \cdot e^{-2sT}, \quad S_2(s) = \frac{(d d_c) s + d k_c}{(2d_c + d) s + 2 k_c}.
\]  

(3.1)

The pole locations of both systems \( S_1 \) and \( S_2 \) are dependent on the system characteristics, and the real part is guaranteed negative for any positive values of \((m, d, k_c, d_c)\) – meaning algebraic stability is guaranteed. Since the delay operator itself is also stable, this means that conditions 2 and 3 are satisfied. Next, observe that the relative degree of \( S_2 \) is zero – making it a positive real function. However, owing to the delay operator, the Nyquist plot of \( S_1 \) is not strictly in the right-half plane, and therefore is \( S_1 \) not strictly positive real, and condition 1 is not satisfied. Thus, stability cannot be guaranteed for any non-zero delay.

3.6 Simulation

To evaluate the performance of the different configurations, simulated implementations of each configuration are evaluated and compared against each other. The client and the observer each have a dynamic model corresponding to the benchmark setup shown in Chapter 2, with each model containing half the friction \( d \), and with the entire mass \( m \) at the client side. A step-input of force from the user is applied, given by the equation

\[ F(t) = \theta(t) \cdot 10 \text{N}, \]

where \( \theta(t) \) is the dimensionless unit step function, whose value is 1 for non-negative arguments, and 0 otherwise.
The controller remains on the same form, regardless of configuration, and is given by the transfer function

\[ C(s) = \frac{d_c s + k_c}{s}. \]

The numerical values for the parameters are borrowed from [21], and can be seen in Table 3.2. The proportional gain has been tuned to be less aggressive by a factor of 3, since not all configurations provide a stable simulation result for the original value.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass</td>
<td>( m )</td>
<td>( 2.5 \times 10^{-1} ) kg</td>
<td></td>
</tr>
<tr>
<td>Damping</td>
<td>( d )</td>
<td>( 2.5 \times 10^2 ) Ns/m</td>
<td></td>
</tr>
<tr>
<td>VC stiffness</td>
<td>( k_c )</td>
<td>( 2.0 \times 10^3 ) N/m</td>
<td></td>
</tr>
<tr>
<td>VC damping</td>
<td>( d_c )</td>
<td>( 3.0 \times 10^2 ) Ns/m</td>
<td></td>
</tr>
</tbody>
</table>

3.7 Results

Simulation results can be seen in Figure 3.5. The characteristic behaviour of each distribution can be seen most clearly in Figure 3.5a, which shows the distinct oscillations caused by a 20 ms delay, with only one client and observer.

A significant difference can be seen when comparing the observer-side controller with the other configurations. In the former, the controller is on the observer-side, so the user only interacts with half the friction and damping, bringing it up to twice the desired terminal velocity before the correction signal has time to return. In the client-controller case, the controller attempts to correct the client-side model back to the origin from the very start, which mitigates the overshoot considerably.

It can also be seen that the steady-state velocity settles at the ideal value, except in the case of the dual controller configuration. This particular configuration remains stable for large delays, but a steady-state error is introduced, caused by energy dissipation in the controllers. Figure 3.5b exaggerates this delay-proportional loss in the case of \( T = 50 \) ms.

Next, in Figures 3.5c and 3.5d, a second client is introduced, with damping distributed evenly between each. The only configuration that remains stable for delays above 5 ms is the velocity-velocity. The figures show the behaviour of the system on the border of stability. It can also be seen that the velocity approaches three times the value of the ideal terminal velocity before converging to the desired value.

3.8 Discussion

To summarise, the overall architecture has some fundamental limitations that lead to unsatisfactory performance. Most importantly is that stability is not guaranteed in the presence
Figure 3.5: Simulation results for the proposed architecture showing the step responses from the different controller configurations under otherwise identical conditions.

of communication delay. It is stressed, however, that this does not mean that instability is guaranteed, but rather that some combinations of parameters and controller gains may render some system signals unbounded. Secondly, because of how the damping is distributed evenly between clients and the observer, large delays may lead to the object’s resistance appearing too low – degrading the user experience. This is more noticeable in the configuration with the controller at the observer-side.

Based on the simulation results, the only realistic choice for an actual collaborative application would be the dual symmetric architecture, since it is the only one that deals with multiple clients at delays that can be expected from internet conditions.

However, there are clear drawbacks with employing this configuration, most notably with regard to transparency. The reason being that the user still only manipulates a fraction of the friction, causing the object to feel too light. This can be compensated by adjusting the controller gain, but this requires manual tuning for some given mass and damping. Furthermore, the delay-proportional steady-state error also causes problems with respect
to the transparency and the consistency, which could further degrade the user experience.
Lastly, as shown in the beginning of the chapter, stability cannot be guaranteed for any arbitrary amount of communication delay. For these reasons, the method will be refined further, and the next chapter explores an alternative way to communicate the necessary information between clients and the observer.
Wave-based communication

As was shown in the previous chapter, the proposed client-observer architecture suffers from a range of issues, with transparency loss and possibly instability as potential outcomes. In this chapter, instead of transmitting power variables (force, velocity) over the network, an alternative approach is proposed where the information is transformed to waves before being transmitted, with the benefit of preserving passivity.

The overarching structure will be the same star topology described in Chapter 3, with the observer in the center. Each client retains the local model, which exists in the power space. The only change is that all incoming and outgoing communications take the form of waves, travelling along an imaginary physical medium.

To understand how the problem can be transformed to the wave space, it is useful to first consider only the case of one virtual object. The purpose of the control architecture here is to communicate the object’s absolute position to the central observer. This is solved in the power space implementation in Chapter 3 by either transmitting the position over the network, or the force needed to move the observer’s object to this position. As shown in Chapter 3, the results did not satisfy the stability and transparency requirements.

The underlying reason for the stability loss is the phase shift caused by the delay operation – which describes a non-physical propagation of information. According to relativistic mechanics, energy and matter cannot be transported from one point of space to another instantaneously. Instead they need to travel for some non-zero time in the form of waves along some physical medium. Modelling information flow as waves allows the passivity criterion to be satisfied, since the medium does not create energy by itself.

4.1 Transmission line model

To realise the above mentioned test case in the wave space, one can imagine the network as a uniform transmission line with delay $T$ and characteristic impedance $Z_c$, with two restrictors $(R_1, R_2)$, one in each end, see Figure 4.1.
The network is the transmission line, the flow in this case represents the velocities of the two different models, and the pressure represents the effort that is applied to the models at each end of the network. The restrictors are energy-dissipating elements, and represent the friction acting on the models, at each end. In this example the mass has not yet been considered, which means that steady-state is attained instantly, without any transient behaviour.

The flow through the restrictors is determined by the pressure difference and the resistance, written as

\[ q_2(t) = \frac{p_1(t) - p_2(t)}{R_1}, \quad (4.1) \]
\[ q_3(t) = \frac{p_3(t) - p_4(t)}{R_2}, \quad (4.2) \]

and the directions of the flow are defined such that

\[ q_1(t) = -q_2(t), \quad (4.3) \]
\[ q_3(t) = -q_4(t). \quad (4.4) \]

Lastly, the flow will travel down the transmission line, modelled as

\[ p_2(t) = Z_c q_2(t) + c_2(t), \quad (4.5) \]
\[ p_3(t) = Z_c q_3(t) + c_3(t), \quad (4.6) \]

with wave characteristics

\[ c_2(t) = Z_c q_3(t - T) + p_3(t - T), \quad (4.7) \]
\[ c_3(t) = Z_c q_2(t - T) + p_2(t - T). \quad (4.8) \]

Consider the pressures at the ends of the system, \((p_1, p_4)\) the input to the system, and the flows \(q_2, q_4\) at the two ends the output. Combining the above equations yields the two equations for the flow at the two ends

\[ q_2(t) = \frac{R_2 + Z_c}{Z_R^2} p_1(t) - \frac{2Z_c}{Z_R^2} p_4(t - T) - \frac{(R_2 - Z_c)}{Z_R^2} p_1(t - 2T) + \frac{(R_1 - Z_c)(R_2 - Z_c)}{Z_R^2} q_2(t - 2T) \]
\[ q_4(t) = \frac{2Z_c}{Z_R^2} p_1(t - T) - \frac{R_1 + Z_c}{Z_R^2} p_4(t) + \frac{(R_1 - Z_c)}{Z_R^2} p_4(t - 2T) + \frac{(R_1 - Z_c)(R_2 - Z_c)}{Z_R^2} q_4(t - 2T) \quad (4.9) \]

where

\[ Z_R = \sqrt{(R_1 + Z_c)(R_2 + Z_c)}. \]
4.2 Choice of design parameters

In the given transmission line model, the network delay $T$ is the only parameter that cannot be chosen freely. The remaining three parameters, $Z_c$, $R_1$ and $R_2$ are design variables and can be chosen to shape the system dynamics. To determine how the characteristics of the system should be designed, it will prove useful to study how the different parameters impact the system behaviour.

Looking at Equations (4.9), (4.10) it can be seen that there will be some transient behaviour, which will be dependent on the delay $T$, as well as the other characteristics of the system.

4.2.1 Steady-state analysis

Defining steady-state as the system state at some $t \geq t_\infty$, for which $x(t_\infty) = x(t_\infty + \Delta t)$, where $x(t)$ represents all system signals, and $\Delta t$ some arbitrary, non-negative time increment. It follows that for all system signals in steady-state, it must hold that $x(t) = x(t + T)$, which gives the steady state outputs

$$q_2(t) = q_4(t) = \frac{p_1(t) - p_4(t)}{R_1 + R_2}, \quad \forall t \geq t_\infty \quad (4.11)$$

It can also be seen that with inputs equal to zero for all $t < 0$, for any input at $t = 0$, the time-delayed terms of Equations (4.9), (4.10) are equal to zero for all $0 \leq t < T$. This means that the outputs are given by

$$q_2(t) = \frac{p_1(t)}{R_1 + Z_c}, \quad \forall t : 0 \leq t < T \quad (4.12)$$

$$q_4(t) = \frac{-p_4(t)}{R_2 + Z_c}, \quad \forall t : 0 \leq t < T \quad (4.13)$$

Looking at Equations (4.11), (4.12), (4.13), the evolution of the states over time can be described as follows. After step-inputs at $t = 0$, the initial conditions $q_2(0)$ and $q_4(0)$, will be described by the resistance at that respective end, and the characteristic impedance. At intervals of length $T$, the states will change with discrete steps, approaching the final steady-state value, which itself is independent of both $T$ and $Z_c$.

4.2.2 Anechoic termination

It becomes apparent that the most trivial choice of parameters that provides the best possible transfer behaviour is to choose $Z_c = R_1 = R_2 = R$. This makes the initial conditions coincide with the steady state, without any transient behaviour. In fact, looking at Equations (4.9), (4.10) again, the higher-order terms are effectively cancelled by this choice, which renders the system equations as follows.

$$q_2(t) = \frac{p_1(t) - p_4(t - T)}{2R} \quad (4.14)$$

$$q_4(t) = \frac{p_1(t - T) - p_4(t)}{2R} \quad (4.15)$$
For this configuration, the steady-state value is attained instantly, and the equivalent pressures are delayed at each end by a factor of $T$. This transfer behaviour, produced by this choice of parameters, is called anechoic termination \[35\], meaning “non-echoing”, or “echo-free”. It can be compared to critical damping in an oscillating system, in that it provides the fastest possible response without any overshoot or oscillations.

Although it will not be shown analytically, it is found through experimentation that for $N$ number of clients with identical resistance $R$, anechoic termination happens for

$$Z_c = N R. \quad (4.16)$$

Finally, it can be shown that each client experiences a total impedance given by

$$d_{\text{felt}} = R + Z_c = R (N + 1).$$

To provide the user a transparent feeling, the resistance and impedances should be chosen so that $R = d / (N + 1)$, which produces the total mechanical damping $d$ that the user expects to feel.

### 4.2.3 Object mass distribution

The masses are inductive elements, which are connected in series with the resistor, meaning that the same flow passes through them as the resistors. Inductance is an energy-storing component, which stores energy as flow. When a step-like change in effort is applied at one terminal, the induction will give rise to a transient, after which the same steady-state flow is reached, as though the inductance was not there.

Now consider the augmented transmission line, with inductors $L_1$ and $L_2$ on each end of the line. This behaviour is described by the differential equations

$$p_1(t) = p_0(t) - L_1 \dot{q}_2(t), \quad (4.17)$$
$$p_5(t) = p_4(t) - L_2 \dot{q}_4(t), \quad (4.18)$$

which, when combined with Equations (4.14) and (4.15), gives

$$\dot{q}_2(t) = -\frac{2 Z_c}{L_1} q_2(t) + \frac{1}{L_1} p_0(t) - \frac{1}{L_1} p_5(t - T) - \frac{L_2}{L_1} \dot{q}_4(t - T) \quad (4.19)$$
$$\dot{q}_4(t) = -\frac{2 Z_c}{L_2} q_4(t) + \frac{1}{L_2} p_0(t - T) - \frac{1}{L_2} p_5(t) - \frac{L_1}{L_2} \dot{q}_2(t - T) \quad (4.20)$$

for the anechoic case.

It can be seen that the inductance term on the opposite side of an applied pressure gives rise to an asymmetric pressure wave, proportional to the rate of change, which will reflect back with a period of $T$.

Since it is desirable for the observer-side model to reach consensus as quickly as possible, and the presence of wave reflections is undesirable, a straightforward solution is to make that object massless ($L_2 = 0$), and endow the client’s object with the entire mass, ($L_1 = m$). This is similar to the design choices made in Chapter 3, but also resolves the issue where the object’s resistance is lower than what it should be.
4.3 Implementation

Considering the controller configurations from Chapter 3, the use of wave variables allows for a fourth variation. As stated in [38], “designing the system entirely in wave space removes the need for both master and slave controllers”. Considering that the transform renders the communication link as a spring, the force-force configuration can also be implemented, which simplifies the analysis by removing the controllers completely.

Using the force-force configuration as an example, the equations of motion for the observer becomes

\[ d\ddot{x}_0 = \sum_{n \in N} F_n = \sum_{n \in N} \left(\sqrt{2Z_c}u_n - Z_c \dot{x}_0\right) = -N Z_c \dot{x}_0 + \sqrt{2Z_c} \sum_{n \in N} u_n \]  

(4.21)

and for some client \( k \) becomes

\[ m \ddot{x}_n = F_n - d\dot{x}_n - \left(Z_c \dot{x}_n - \sqrt{2Z_c} v_{0,n}\right) \]  

(4.22)

A wave filter is implemented like in [37], since numerical errors give rise to some minor problems. It is not shown here, but it follows intuitively that increasing the aggressiveness of the filter gives smoother results, but less responsive transmission behaviour. In [37], a rule-of-thumb for the bandwidth \( \lambda \) as a function of the delay \( T \) is suggested as

\[ \lambda \approx \frac{1}{T}, \]

but for this application a constant value of \( \lambda = 1000 \text{ rad s}^{-1} \) is chosen.

4.4 Results

Results for one client and massless observer, using all four configurations (Table 3.1) are shown in Figure 4.2. The first observation is that for only one client and the massless observer (Figure 4.2b), the force-force configuration performs with ideal transparency. Comparing Figures 4.2a and 4.2b, it can be seen that all state trajectories converge to the force-force configuration, as the controller gains are increased. Figure 4.3 shows the undesired wave reflections that can arise. In Figure 4.3a, the mass is evenly distributed to the observer and the client, and as a result, some oscillations can be seen, as shown by Equation (4.19). In Figure 4.3b, the observer is massless, but a second client is connected to the observer. The resistances and wave impedances are adjusted to provide anechoic termination, nonetheless, the presence of another client causes severe oscillations, that last several periods. Similar to the previous chapter, the force-velocity is the most susceptible to the oscillations in both cases, though unlike the power-space counterpart, it does not grow unstable.
4.5 Discussion

As mentioned earlier, by increasing the stiffness of the corresponding controllers, the different configurations all converge to the controller-free implementation. For the sake of future tests, the force-force configuration is treated as the baseline. This allows for running further experiments, while completely decoupling the controller from the analysis.

One limitation of the force-force configuration is that numerical errors may lead to diverging positions across clients, without the spring force correcting this. This can be solved by either introducing top-level correcting control as shown in [37], or by transmitting the positions directly, using the wave integral [38].

Wave reflections are still a problem and will be addressed in the next chapter.
Reflection compensation

As shown in the previous chapter, the wave-based communication introduces oscillations in the velocity of the client objects, so called wave reflections. The resulting oscillations will degrade the quality of experience, and should be prevented if possible. In this chapter, a novel solution is proposed to try to mitigate the undesired wave reflections by attempting to estimate and cancel them from the returning wave.

5.1 Problem description

The distributed mass is the underlying cause of wave reflections, which manifest as periodic jumps in velocity. The effect is caused by the stationary masses of the other clients storing kinetic energy, and eliciting a force opposed to the rate of change in velocity of the original client. Figure 5.1 shows the incident and reflected wave at client 1.

Figure 5.1: Wave space analysis of client 1. Since no other clients are performing work, the returning wave should be zero for all time, but the inductance of the second client causes power to flow back.
Studying the figure closely also reveals that the echo bounces back several times, which degrades the experience further.

5.2 Overview

The principle of the reflection compensation is to estimate the undesired reflection using model-based prediction, and subtract it from the returning wave when it arrives, while a top-layer modulation algorithm ensures that the passivity condition is never violated. The algorithm proposed in this section can be used for any linear dynamics, and is generalised for any desired degrees-of-freedom. For the sake of the analysis, all dynamical models will be assumed to be complex-frequency impedance matrices \( Z(s) \), where the independent variable will be dropped for compact notation. Dimensionless transfer functions, such as that between wave variables, will be denoted by the letter \( G(s) \).

For completeness, the design choices from the previous chapters will not be inherently assumed. This means that the client impedances \( Z_c \) and any other local parameters (bandwidth, delays, etc) are assumed constant, but not identical across clients. A subscript index will be used to differentiate them from each other. The observer-side model is also denoted by the complex impedance \( Z_0 \), and other superscript notations will be used to describe the different stages of the wave propagation.

Lastly, since the characteristic impedance of each transmission line is strictly a design parameter, it is assumed to be identical across all clients, and is given by the transfer matrix \( Z_c \). Naturally, the proposed solution suffers from some inherent weaknesses, which will be addressed in the discussion-section at the end of this chapter.

5.3 Identifying the transfer function

Since all impedances are dictated by the dynamics engine, every step of the wave propagation is known, including the values for the physical parameters. To create a model for the estimation, the transfer function from the incident wave to the reflected wave is identified by observing the propagation. Considering the block diagram in Figure 3.4

First, consider client \( n \) with \( n \in \mathbb{N} \), being moved by a reflected wave coming off the observer. This is given by

\[
G_{cw}^n = \sqrt{2} Z_c (Z_c + Z_c)^{-1} \sqrt{2} Z_c - I. \tag{5.1}
\]

Assuming that each client has a wave filter, each wave must travel through the filter with bandwidth \( \lambda \), and the estimated time-delay \( \hat{T} \). The transfer function is given by

\[
G_{fd}^n = \frac{\lambda_n}{s + \lambda_n} e^{-s \hat{T}_n}. \tag{5.2}
\]

The overall impedance of one client, as seen by the observer, is then given by

\[
Z_{co}^n = -\sqrt{2} Z_c (G_{fd}^n G_{cw}^n + I)^{-1} \sqrt{2} Z_c + Z_c. \tag{5.3}
\]

The total impedance of the observer plus all attached clients is then given by the sum

\[
Z_{oc} = Z_0 - \sum_{m \in \mathbb{N} \setminus \{n\}} Z_{co}^n, \tag{5.4}
\]
note that the sum operator has no summands in the case of a single client.

Finally, the complete wave-reflection transfer function from the incident wave to the reflected wave is given by

$$G_{wn}^{wr} = \left( \sqrt{2} Z_c (Z_{oc}^{nc} - Z_c)^{-1} \sqrt{2} Z_c + 1 \right)^{-1}.$$  \hspace{1cm} (5.5)

Again, consider the special case of only one client, where the total observer impedance is equal to the observer impedance only. In the anechoic case with a massless observer, it holds that $Z_0 = Z_c$, and (5.5) becomes undefined, since no reflections occur in this case. While the transfer function becomes zero in the SISO-case, the matrix inversions make the expression undefined otherwise.

### 5.4 Passivity modulation

Since the estimation-based compensation will alter the numerical value of the incoming wave with a correction term, additional power may be introduced into the system, which would violate the condition for passivity, invalidating any guarantees for stability. To tackle this, a modulation of the correction term is proposed. The idea is inspired by the position feedback algorithm in [37], which respects passivity by “bounding the magnitude of the corrected wave command by the original uncorrected version”, and shares the same logic.

The incoming reflected wave is already denoted as $v(t)$. The estimated echo from the model is then considered the correction term, $\hat{v}(t)$. Next, let the compensated wave be defined as $\tilde{v}(t) = v(t) - \hat{v}(t)$. Without going into the actual implementation, it should be stated that the modulation algorithm is applied to the correction term. A simplified but equivalent definition using the corrected wave $\tilde{\varphi}(t)$ that is fed back into the inverse wave transform is given by the following if-then statement

$$\tilde{\varphi}^*(t) = \begin{cases} 
\hat{v}(t), & \text{if } |\hat{v}(t)| \leq |v(t)| \\
|v(t)| \cdot \text{sgn} \hat{v}(t), & \text{otherwise}
\end{cases}.$$  \hspace{1cm} (5.6)

A block diagram of the entire process, including the estimation and modulation can be seen in Figure 5.2.

![Figure 5.2: Block diagram of the augmented process, including compensation](image)
5.5 Results

Unless specified, all simulations assume that all communications consist of the same constant delay. The delay is also assumed to be known, so that \( T = \hat{T} \). Figure 5.3a shows the system response of the dual-controller configuration, after a step-input of force at client 1, with the same system parameters as in Chapter 3, and without any compensation. Figure 5.3b shows the step response of the same conditions, but with active compensation enabled. The dot-dashed and darker line represent the velocity of the second client.

(a) Uncompensated force-force configuration    (b) Compensated force-force configuration

Figure 5.3: Step-response comparison with and without compensation, \( T = 20 \text{ ms} \)

As can be seen by comparing the two figures, the largest effect of the compensation algorithm is on the client exerting the force on the shared object. The slight overshoot of the second client side is not caused by any undesired reflections, since it happens as soon as the incident wave reaches it. For this reason it is not eliminated by the compensation algorithm, but appears to disappear after one additional period.

Figures 5.4a and 5.4b show the deviation from the dashed black line for the first client. Moreover, the velocity deviation is reduced from at most 37% to less than a 1% relative error, which corresponds to a few tenths of a millimetre per second.

The compensation scheme works similarly on other kinds of input signals. Next, a sine wave is used as input, the signal taking the form

\[
F(t) = \sin(8\pi t) \cdot 10 \text{ N}.
\]

Figures 5.5a and 5.5b show the system response to the periodic input force, with and without compensation, respectively.

Finally, the effect of uncertainty in the estimation is investigated. Figures 5.6a and 5.6b show the system response with overestimated and underestimated delay times, respectively. Consider the communication delay as modelled by (5.2). For this simulation, the uncertainty of the delay is modelled as \( \hat{T} = \Delta T \), where \( T \) is the actual delay used in the simulation, and \( \Delta \) the multiplicative factor. In the ideal case, it is equal to one. In the sensitivity analysis included here, the scalar used in the underestimated case is \( \Delta = 0.8 \),
Reflection compensation

Figure 5.4: Loss of transparency with and without compensation, $T = 20\,\text{ms}$

(a) Velocity deviation without compensation

(b) Velocity deviation with compensation

Figure 5.5: Sine-wave response comparison with and without compensation, $T = 20\,\text{ms}$

(a) Uncompensated force-force configuration

(b) Compensated force-force configuration

and for the overestimation, it is the reciprocal; $\Delta = 1.25$. While the algorithm no longer matches the reflection perfectly, it still holds that the modulation prevents any increase in energy, so that the system remains stable. More on the drawbacks of this method and some potential workarounds will be discussed in the next section.
Chapter 5

5.6 Discussion

As shown in the previous section, the proposed solution is effective at cancelling undesired wave reflections from wave-based communications in collaborative environments. However, the proposed method suffers from some weaknesses. Although the dynamics are considered to be linear and time invariant for the benchmarking purposes, in practice, this is not always the case. In the higher-dimensional case, even if the dynamics are linear, it might be the case that some parameters may change during runtime. Consider, for example, a case where the user is free to pick up the VO. During simulation, the frictional coefficient will change as the object leaves the ground. Even if the change in physical parameters is communicated between instances, during the time it takes for the information to travel, incorrect estimations could lead to unwanted behaviour. In [40], the issue of updating a local model within a teleoperation setting is addressed.

For the sake of this benchmarking test, it is assumed that the physical properties of the system do not change. Nonetheless, a few issues must still be addressed. The only uncertainty in the estimated parameters for this case would be the communication delay and the filter bandwidth \( \lambda \). Over an internet connection, the communication will be subject to more severe conditions, such as jitter (variable delay), unordered or lost packets. For the sake of simulation, the delay is assumed constant and equal between clients, but the compensation scheme does support different values for different clients. For an actual implementation, the observer would likely have to keep a record of each individual average communication delay and transmit this information to each client on a regular basis.

One alternative implementation would involve placing the reflection compensation on the observer-side. Since the reflected wave passes through it, the observer is just as suitable at predicting what part of the wave is undesired. This has several merits, but primarily, since the observer has less transport delay to each other client, and is assumed to run on more powerful architecture, it would likely be better suited for running more advanced model updating algorithms. This variation will not be investigated further within the scope of this thesis, but will be detailed more in Chapter 7.

(a) Over-estimation, \( \hat{T} = 25 \text{ ms} \)

(b) Under-estimation, \( \hat{T} = 16 \text{ ms} \)

Figure 5.6: Sensitivity test of compensation algorithm, \( T = 20 \text{ ms} \)
Lastly, it should be mentioned that even in the case where $\hat{T} = T$, the virtual object still shows small oscillations in velocity. The cause for the persistence of the error is a slight model mismatch, whose underlying reason in turn is likely numerical and originates from how the transfer function is calculated. Since the transfer function involves delays at different stages of the wave propagation, the system is handled in the solver as a state-space system, which differs from the system of differential algebraic equations running as the main simulation.

In an actual implementation, the bandwidth of the wave filter would likely be decreased, which would smoothen the dynamic behaviour, but lead to less responsive remote interactions.
Performance comparison

In this chapter, the complete proposed solution is evaluated side-by-side with more conventional configurations. The final design consists of the client-observer architecture, with wave variable communication, and active reflection compensation at each client, with no uncertainty in the estimation. Each evaluated architecture will consist of two users, interacting with the same object over a network of constant and equal delay. The analysis is qualitative, and involves studying the time-evolution of the velocity of the object at each instance.

6.1 Conventional structures

The first architecture that will be evaluated is the classic, centralised CS-state architecture, which utilises the server for all dynamics simulation, taking only the sum of the delayed forces from each user as the input.

Next, a state-of-the-art, distributed P2P architecture is tested. The implementation is the virtual coupling scheme 1 from [20] with one significant modification. Similar to the hybrid architecture, each peer contains a model containing the entire mass. The architecture utilises a dual-symmetric controller directly between each peer, and the parameters have been tuned by hand to give a stable result with desirable dynamic properties.

6.2 Results

The architectures will be evaluated via two experiments; one involving input from only one client, and the second involving inputs from both clients simultaneously.

6.2.1 Unilateral interaction

Figure 6.1 shows the step response of each respective architecture. The full line shows the client/peer at which the force is applied, and the darker, dot-dashed line shows the reacting client. The force from user 1 is given by

\[ F_1(t) = \theta(t) \cdot 10 \text{ N}, \]
where \( \theta(t) \) is the dimensionless unit step function, whose value is 1 for non-negative arguments, and 0 otherwise.

The time-domain behaviour for each client will be categorised into three headings. First, the response time, which is the time it takes for the object to start moving after the force has been applied. Next, the time constant, which is determined by the characteristic slope of the step response curve. While wave filters might have a small effect on the slope of the curve, the dominant effect comes from the inductance of the virtual object, in this case, the mass. For the sake of comparison, the time constant is calculated as \( \tau = m/d \), i.e. the quotient of the mass and damping. Finally any other transient phenomena such as overshoot or similar will be grouped in the third category, namely transients. Table 6.1 shows a summary of the categorised output of the tested architectures.

<table>
<thead>
<tr>
<th></th>
<th>Response time</th>
<th>Time constant</th>
<th>Transients</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Client 1</td>
<td>Client 2</td>
<td>Client 1</td>
</tr>
<tr>
<td>CS</td>
<td>2T</td>
<td>2T</td>
<td>( \tau )</td>
</tr>
<tr>
<td>P2P</td>
<td>0</td>
<td>T</td>
<td>( \tau )</td>
</tr>
<tr>
<td>Hyb</td>
<td>0</td>
<td>2T</td>
<td>( \tau )</td>
</tr>
</tbody>
</table>

As expected, the client/server setup produces identical results for both clients, since they are reflections of the same simulated object. The response time for both clients is given by the round-trip time (RTT) of the system, which is the time it takes for the information to reach the server and come back. In this case it is the sum of both the transport delays, \( \text{RTT} = 2T \). The time constant is the same as the ideal case, and equal for both clients, but as shown in [11] and [12] among others, the effect of the delay alters the experienced inductive behaviour, which negatively impairs the QoE for the local user.

Next, consider the peer-to-peer setup. Since the rendering is done locally, the response time of the local client is zero, and because the model contains the entire mass, the time constant is equal to that of the ideal case. From its decentralised nature, the response time of the second client is determined directly by the delay, and because each object has the entire mass, the time constant is twice that of the ideal case. Furthermore, persistent oscillations caused by reflections are present throughout, and take considerable time to dissipate. Additionally, the steady-state error in velocity first described in Chapter 3 is present here.

Finally, as expected, the hybrid architecture shows some of the strengths of both architectures, and some of their weaknesses. Like in the P2P setup, the local interaction is as responsive as in the ideal case, but also free from wave reflections because of the active compensation. The drawbacks appear for the remote user, where like in the CS-case, the communication delay consists of the sum of both transport delays, and like in the P2P case, the increased mass leads to a larger time constant. As noted earlier, a small overshoot is also present for the remote user, but vanishes after one period of the RTT.
6.2.2 Bilateral interaction

Next, the performance of the respective architectures are evaluated for a collaborative application, where both users are interacting directly with the object. In this experiment both users apply an equal but of opposite direction force to the object. When after $t = 0.15\text{ s}$ the second user stops applying any force, the first user’s input remains unchanged. The input signals are given by

\[
F_1(t) = \theta(t) \cdot 10\text{ N} \\
F_2(t) = (\theta(t) - \theta(t - 0.15)) \cdot (-10\text{ N}).
\]

For a visualisation of the input signals, see Figure 6.2.

![Figure 6.2: Comparison](image)

A comparison between the response of the different architectures is shown in Figure 6.3. In the CS case, the forces cancel out at the server, since the delay is equal at both clients. In the two distributed cases, the local dynamics engine causes the virtual objects at each instance to diverge, before the respective information has travelled across the transmission line. As seen before, this happens faster for the P2P case, since the total delay is smaller. The oscillations are present in this case too, with larger magnitude than previously observed.

An interesting phenomenon that has previously been unobserved is the undershoot that occurs in client 2 on the hybrid architecture, at $t = 0.17\text{ s}$, which is one RTT after the input at client 2 ceases. It is a typical wave reflection, but it does not get compensated by the estimation scheme. The explanation is as follows. The force from client 1 comes to client 2 in the form of a wave, with positive contribution to the overall power. When the force from user 2 becomes zero, the step-like change in power is negative. This change in power travels along the transmission line, and is reflected on the other client. When it returns it results in a reduction of the power transmitted from client 1, which in turn leads to the drop in velocity that can be seen. In order to correct this effect, the compensation scheme would have to add a positive correction term to the total power, which would violate the passivity condition.

This phenomenon, along with some other observations are discussed further in the next section.
6.3 Discussion

To summarise, the dynamic behaviour of the virtual object in the hybrid architecture shows some common traits with both conventional architectures, and can be said to adhere to its hybrid nature. As mentioned previously, the hybrid architecture is intended to capture the strengths of both architectures, but as such, it also suffers from some of the weaknesses from both.

Besides the dynamics of the virtual objects, there are other factors to consider when designing the controller architecture. Section 7.2 contains a more comprehensive discussion about issues such as scalability, hardware, practical limitations and use-cases.

Finally, regarding the uncompensated wave reflections described in the previous section, it is no coincidence that this phenomenon appears first when considering simultaneous inputs from multiple users. Only in situations where power is flowing from one user to another can reflections decrease the overall power transmitted. As described earlier, the wave reflection remains uncompensated in order to retain the guarantee for passivity. There are likely to be certain circumstances where the compensation would be safe to do, and although it will not be pursued further within the scope of this thesis, it could be investigated in the future.
Figure 6.1: Simulation results showing the step responses from different distributed control architectures using the same benchmarking setup, with client 1 applying force, and using a constant delay of $T = 10\,\text{ms}$. 

(a) Client/server with dynamics engine only on the central node  

(b) Peer-to-peer setup with a manually tuned dual-symmetric controller configuration  

(c) The proposed hybrid, wave-based architecture with wave reflection compensation
Figure 6.3: Simulation results showing the responses from different distributed control architectures using the same benchmarking setup, with both clients applying force, and using a constant delay of $T = 10\,\text{ms}$. 

(a) Client/server with dynamics engine only on the central node

(b) Peer-to-peer setup with a manually tuned dual-symmetric controller configuration

(c) The proposed hybrid, wave-based architecture with wave reflection compensation
Conclusion

To conclude the thesis, this chapter will reflect on the questions formulated in Section 1.3.1, and summarise the findings. The first question is in regard to the optimal configuration and parameter choice, when considering the system stability, consistency, and transparency.

In order to even approach the question of consistency or transparency, stability must first be assured. As shown in Chapter 3, the introduction of any nonzero communication delay between client and observer causes the guarantee for stability to be lost. This assumes communicating in the power-space, and is true even if all interconnected subsystems are by themselves, stable. The problem is solved by instead communicating in the wave-space, as presented in Chapter 4. This modified structure preserves passivity across time-delayed communication, and thus assures BIBO-stability. There are other ways to enforce the passivity of a distributed system, but these methods are not evaluated within the scope of this thesis. A consequence of the wave-based communication is that an external controller becomes redundant, since the transmission line acts as the controller.

The other side to the first question considers the optimal distribution of the design parameters. The definition of optimality in this application is not ubiquitous, and does not rely on a rigorous mathematical definition. Instead, a theoretical examination of the time-domain behaviour, coupled with a qualitative assessment of simulation results, is used to determine the overall trend of different strategies. First, with the introduction of the transmission line modelling, comes the need to design the characteristic impedance. It is shown that anechoic termination can be attained, while also shaping the virtual object’s damping coefficient. This has the result that each user experiences the correct amount of friction, and no unwanted oscillations. Next, it is determined that the observer’s object need not contain any mass, since this only introduces an inductive effect on the other side of the transmission line, which causes echoes. Additionally, this also means that in the case with a single user, the observer object reproduces the client’s object exact motion with a delay equal to the RTT. Subsequently, the virtual object’s mass is distributed such that each client contains the full mass, which gives a more transparent feeling for local interactions, but reinforces the undesired wave reflections. While no “delay compensation” as mentioned in the research question, is explored, a compensation scheme is proposed in Chapter 5 to deal with the reflections that arise during collaboration.
Finally, while the second question regarding the performance comparison is already discussed in-depth at the end of Chapter 6, it is worth mentioning that the proposed solution shows promising results, and that there is reason to continue the research.

7.1 Summary

To summarise, the basis for the contribution put forward in the thesis is proposal of a novel, distributed architecture for transparent interactions in collaborative haptic virtual environments. The most important contribution is the introduction of a dynamics engine at the clients in a centralised environment, and a repurposing of the server as a centralised observer for maintaining consistency. This gives each user a locally responsive experience, and does not rely on centralised rendering to interact with the virtual object.

Another important notion is that the centralised object acts as a flexible joint element, with each client attached through a transmission line, acting like a virtual coupling. This paradigm also ensures passivity, and a theorem is provided in Chapter 3 which gives the requirements for system stability. Passivity of all subsystems is a central concept in this theorem.

The largest drawback of the architecture is the undesired wave reflections that come as a side-effect of the wave-based communications. A proposed model-predictive compensation scheme running at each client is shown to eliminate the biggest effect of these reflections, but it is sensitive to uncertainty and other conditions, and should be revised for use in more complex collaborative virtual environments.

Finally, the feasibility of the wave-based architecture is shown in a 1-DOF, linear use-case involving multiple users, and the overall performance is compared to conventional architectures. Compared to a peer-to-peer configuration, the proposed architecture provides a local responsiveness closer to the ideal case, while still retaining the benefits of the centralised topology.

7.2 Discussion

While the definition of collaboration as used within the scope of this thesis covers the requirements on simultaneous manipulation of shared virtual objects, one thing that remains to be addressed is the number of users, and the scalability of the system. In the state-of-the-art, it is generally the case that applications utilising a P2P architecture will only consider two users [16], [20], and this also holds for the case with wave variables [21].

A large benefit with the proposed architecture is the scalability that it offers. In the centralised topology, each client only communicates with the observer, regardless of the amount of clients connected. The computational burden remains unchanged for the connected clients, with the exception of the wave reflection estimation. This issue is addressed in the next section, where a reimplementation is suggested.

For a pure P2P setup with $N$ peers, each peer would have to implement $N - 1$ communication pathways. Any controller interaction would not only travel along the intended pathway, but also indirectly through each other peer, which could cause undesired dynamic behaviour. In comparison, each client in the hybrid solution is largely unaffected by the number of users, only a few parameters need to be reconfigured each time a client
conclusion joins or leaves. Most importantly, each client only maintains one bidirectional, wave-based connection with the observer. For the sake of the argument, it would be possible to remove the observer, in the special case that the system only support a maximum of two users, since the nature of the waves are entirely modular. However, if the compensation scheme was moved to the observer prior to this, the system would suffer from the reflections.

While the architecture scales well in regard to the clients, the same can not be said about the observer, and it follows that hardware and other practical limitations will need to be considered. By moving the prediction to the observer, the workload for the observer scales linearly with the user count. A dedicated server hardware running the observer software would have to be dimensioned to take into consideration the expected amount of users. As mentioned earlier, the required refresh rate for a haptic system is high, compared to other sampled systems, with a required rate in the order of kHz. Different levels of complexity could be considered for the prediction algorithm, which could improve the scalability of the observer as well.

7.3 Future work

The proposed wave-based architecture has been tested in a simplified simulation environment as a proof of concept, but has yet to be evaluated on hardware, and subjected to actual internet conditions. Before the architecture can be tested on a real distributed haptic virtual environment, a few modifications and improvements must be made.

First of all, as explained in Chapter 4, all simulations in the subsequent chapters are run without any explicit controllers, even though the transmission line can be seen a virtual coupling. This is conscious, as it allows the evaluation of the network architecture without considering the impact of controller gains, and eliminates the need for controller tuning in this thesis. That said, in an actual implementation, other network conditions will be present, such as jitter and packet loss. Since only the velocity is transmitted and integrated locally at each instance, disruptions in the communication can lead to divergent states. In this situation, it is reasonable to believe that the simulation instances can drift apart, and the application will require some position feedback control.

As shown in Chapter 4, the most desirable controller behaviour comes from the observer-side controller configuration. To get position control, the wave integrals presented in the Preliminaries could be used, but a more sophisticated solution may involve a slightly different approach, and will be elaborated on in the next section.

Before concluding, one last issue must also be addressed. As brought up in the discussion section of Chapter 6, the reflection compensation algorithm would perform better if it was migrated to the observer. The compensation would then sit on the right side of the delay, if Figure 5.2 is used as a reference. This has many benefits; to begin with, it would eliminate the need for estimating the delay and filter between the directly affected client and the observer. Next, it would give a better estimation of the delay between the observer and the other clients, since the observer could calculate this directly. Finally, in a fully 6-DOF system with rotation involved, the dynamics would no longer be linear. If each client would transmit their most recent pose to the observer, this could be used to generate a more accurate estimate of the client dynamics. In this scenario, the previously suggested notion of feedback control becomes important.
7.3.1 Wave-based feedback control

In essence, the observer could use the received pose information to modify the outgoing wave signal to apply feedback control, while still satisfying the passivity criterion. The solution would be a straightforward implementation of the position feedback pathway proposed in [38], which is where the modulation algorithm utilised for the active reflection compensation is first introduced. The position feedback control could be combined into the same framework, and the outgoing wave would be compensated not only to take into account undesired reflections, but also to adjust for position drift between instances. This would ensure state consistency for non-linear dynamics, while still retaining passivity, and thus stability. Additionally, as mentioned in the final discussion, this would also be another step in the right direction with regards to scalability, as each client would then be oblivious to the amount of other users during runtime.
Bibliography


Appendix

Model initialisation

% Simulation time (sec)
T_sim = 0.25;

% Transport delay (sec)
T = 20e-3;

% Number of clients
N = 1;

% Input force (N)
F_m = 10;

%% Virtual environment
m = 0.25; %[kg]
b_d = 2.5e2; %[Ns/m]
d = b_d/(N+1);

%% Observer impedance
Zo = [m d];

%% Virtual coupling
k_T = 2e3; %[N/m]
b_T = 3e2; %[Ns/m]

%% Wave characteristics

% Filter enable
obsFilter = 0;
clFilter = 0;

% Filter cut-off
k = min(1/T,1000);
k = 1000;

% Impedance
b = d*N;

% Wave integral filter coeff
f = 10;

%% Reflection compensation
% Uncertainty
if ~exist('T_hat')
    T_hat = T;
end

% Model impedance
Z = tf([m d],1);

% Client filter and transport delay
H_c = exp(tf('s')*-2*T_hat);
F_c = zpk([],-k,k);

% Client model and wave transform
G_cw = minreal( -(Z - b)/(Z + b) );

% Observer-side wave transform
G_wcw = minreal( b * (G_cw*H_c*F_c - 1)/(G_cw*H_c*F_c + 1) );

% Observer model and other clients
Y_o = minreal( (d - (N-1)*G_wcw )^-1 );

% Complete transfer function
G_refl = H_c*F_c * minreal(ss(Y_o*b - 1)/(Y_o*b + 1));
Figure 1: Block diagram showing the Simulink implementation of the client/observer architecture simulation. Each subsystem representing the client, observer, and network are masked by a `For Each`-block, which allows the number of users to be defined using a workspace variable. To realise the different controller configurations, only the subsystems representing the client and observer need to be updated, since the input-output structure remains the same.