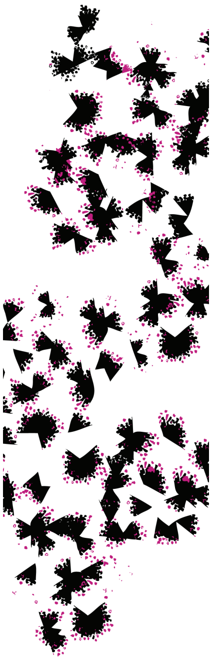
RAM
ROBOTICS
AND
MECHATRONICS



Design of a Reusable Teleoperation Control Architecture Applied to the youBot

S.G. (Sjouke) Rinsma

MSc Report

Committee:
Prof.dr.ir. S. Stramigioli
Dr.ir. J.F. Broenink
D. Dresscher, MSc
Dr.ir. J. van Dijk

Juli 2014

Report nr. 010RAM2014
Robotics and Mechatronics
EE-Math-CS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE.
MIRA
BIOMEDICAL TECHNOLOGY
AND TECHNICAL MEDICINE
CTIT

Abstract

Traditionally, developing a new robotic application means designing a new system completely from scratch, since in most cases no standardized or generic components are available. This is a costly and time consuming task, and one way to speed up this process is by reusing commonly used modules. Unfortunately, a modular robot design is not as intuitive as it may seem, since even small changes in for example the kinematic chain may require large design adjustments. Aiming for a controller design to be usable in dedicated situations, but at the same time applicable in a wide variety of applications, typically is a conflicting requirement. Trying to address an as wide variety of systems, while at the same time providing acceptable performance, requires the design of a robust and reliable controller architecture.

Several initiatives have emerged that aim at serving as a platform for robot designers. The prime objective of the BRICS project is to structure and formalize the robot development process itself and to provide tools, models, and functional libraries, which help accelerate this process significantly. In this MSc theses project, which is in the context of the BRICS project, the main goal is to develop a modular controller architecture for robot controllers, specifically in case of teleoperation. An important aspect is that all controller blocks are designed in such a way that they can be added or removed from the robot control chain without affecting overall stability. This includes designing a generalized interface strategy that applies to all elements in the controller chain.

Firstly, an appropriate input/output strategy has been developed and applied to several control blocks including; Cartesian control, end-effector positioning and interaction tasks. These blocks are designed in 20-Sim using bond graph theory and, in order to verify functionality, the controllers have been implemented on the KUKA youBot setup in a simulation environment. The youBot itself is a mobile manipulator which consists of a mobile platform with a six-DOF robot arm. An important notion in controller design for bilateral teleoperation is passivity, since by demanding passivity of the system overall system stability can be guaranteed. Based on this, the focus has been to use passive control strategies; i.e. wave variable control and IPC (Intrinsically Passive Control).

Simulations have shown that even when using passive controllers, instability can still be a concern. The main cause for this has been due to the discrete nature of signals in the system, which limits the frequency bandwidth allowed. Also, using wave variables, accuracy becomes an important topic, and without additional control, accuracy of the system is quickly compromised, certainly in case of varying delays. Additionally, the use of the wave variable method as a means for stable communication places limits on flexibility of the design, in the sense that the signals which are transmitted across the line are fixed to wave variables. However, due to the modular design, these elements can be easily removed from the controller design and be replaced. The IPC elements on the other hand do provide a great deal of flexibility, since these modules consist of physically interpretable elements which exist everywhere in control theory and modeling. Also, reuse of the modules is possible, since all control elements have been designed as modules containing basic functionality (i.e. a module still functions without the others). Also, because the wave variable modules perform a mapping from power variables to wave variables, it has not been possible to completely generalize the interfaces at all modules. Nevertheless, all modules, in particular the IPC blocks, provide a good basis for setting up a controller library for teleoperation purposes.

Preface

“Work to live, live to enjoy, and enjoy as much as you can.”

After a long, laborious and rather bumpy road toward obtaining my masters’ degree, the journey finally ends with this work; my MSc report. Though the last couple of years have not always been easy, I have always received much help and encouraging words from close friends and family, for which I am more than grateful. Without your unconditional support and friendship, uplifting talks and putting things into perspective, I would not have had the energy and conviction to keep on going and I would not have been able to finish. People know me for always being a busy individual, engaging in a lot of extracurricular activities, which needless to say; I could not be doing without all of you. Making music and being on the road, playing rock shows with my buddies, which we should definitely continue doing in the future — Having drinks, discussions and philosophical wanderings with close friends, engaging in all sorts of initiatives and creative outbursts — Playing all sorts of sports; biking, swimming, surfing, diving and snowboarding — Dancing at parties and festivals, sharing thoughts, experiences and interests, listening to music while playin’ chess and havin’ another, and chilling out until long after sunset. You know who you are, thanks for all the good times!

It has been quite an endeavor, completing my second college degree, at the same time holding a part time job, first at Demcon, later on with 3T. During these last years, and particularly the last few months, I have received much additional support from people connected to the RaM (Robotics and Mechatronics) group and several other departments at the University of Twente, for which I am particularly thankful. This has helped me get things back on track, regain trust and motivation, and not quit with the finish line in sight. As for the people at the RaM group, I have found a lot of inspiration from the robotics courses and all-round involvement of the ever so passionate professor Stefano Stramigioli. My special gratitude goes out also to Peter Breedveld, an associate professor of the RaM group, for assisting me in my second attempt at completing an individual assignment, even though things did not go as smooth as planned.

With regard to my MSc assignment I would like to thank my initial advisors Robert Wilterdink, and especially Yury Brodskiy, for his continuing support. He has been utmost patient and willing, despite the fact that the value of my works’ results to his PhD thesis decayed along the way. Much appreciation also goes out to Douwe Dresscher for adopting the role as my substitute direct advisor after Yury left, dragging me through the last months of my assignment and being personally concerned with my progress. My gratitude most certainly also goes out to another RaM groups’ associate professor and my primary supervisor Jan Broenink, for his strict but fair criticism, and providing me with additional guidance and tools for successfully completing my assignment. Also, I would like to express thanks to Jolanda, for assisting where necessary and genuinely caring, not only for my own, but everyone’s wellbeing at the group.

With this, I hope to close an eventful chapter, flip its last page, and start an interesting new one.

Sjouke Geert Rinsma
Enschede, april 2014

Contents

1 Introduction

Originating from the Greek prefix *tele*- meaning *at a distance*, teleoperation systems provide a human operator with a means to operate and manipulate objects at remote locations. A teleoperation system usually consists of two interconnected manipulators; a *master* being directly controlled by a human operator and a *slave* that aims to follow the master. Using telerobots (i.e. robots operated over distance) is a practical and safe way to allow humans to work in hazardous, hard to reach and remote locations. Teleoperation systems find use in an ever growing variety of applications ranging from telesurgery, outer space construction, operating in radioactive locations, underwater exploration, and so on. Over the past decades, a lot of effort and research has been put into optimizing these systems, being challenged by some persistent obstacles.

1.1 Bilateral Teleoperation

In (Hokayen and Spong, 2005) the main goals of teleoperation from a control theoretic point of view are defined as *stability* and *telepresence.* For obvious reasons, it is important to maintain stability of the closed-loop system irrespective of the behavior of the operator or the environment. Providing the human operator with a qualitative sense of telepresence is another important aspect since this determines how well the user 'feels connected' to the remote manipulator. The latter is regarded as the *transparency* of the system and relates to the coupling between the environment and the operator. The more instantaneous the response of the robot upon operator action, the more transparent the system is said to be. Creating an as intuitive as possible human-machine interaction is one of the most important desires in human operated telerobotics. Ideally, the robot should feel like an extension of the body to the person operating it. Certainly in situations where the operator and robot are separated by a non negligible distance this becomes a challenging task.

Communication over distance is always impaired by *delays* which can drastically degrade performance of a telemanipulation system, resulting in non-intuitive and obscured (i.e. non transparent) system behavior. In case of unilateral control where the master typically only controls position of the remote manipulator this is characterized by a response delay. Without additional control this would result in cumbersome interaction between operator and manipulator since the user would constantly have to anticipate its response. In case of varying delays this becomes an almost impossible task. By equipping the remote manipulator with (force) sensors which reflect the reaction forces, a *bilateral* teleoperation system is created. Providing the operator with tactile information improves intuitive control by creating a sense of 'physical presence' at the remote location. However, in this case delays can easily lead to instability of the control loop. Additional control is necessary to ensure stable operation while trying to retain an as transparent as possible system.

Another challenge is the robot-environment *interaction*, especially in situations where the robot is desired to operate or manipulate objects. Interacting with unknown surfaces or objects can be an important cause for instability due to unpredictable contact response. Moreover, force control is not meaningful in free space since there is no force response from the robot. Conversely, in case of contact with a surface, force is the dominant variable and position feedback provides little to no additional information. This demands for a control architecture capable of coping with non-contact to contact situations and vice versa. Essentially, the causality of the controller should switch from position controlled to force controlled without compromising stability.

1.2 Problem Statement

Traditionally, developing a new robotic application means designing a new system completely from scratch, since in most cases no standardized or generic components are available. This is a time consuming and costly process, which is an important topic in a fast growing field in which time-to-market is decisive and one of the prime objectives is to reduce development time. One way to speed up the development process is by reusing certain commonly used (control) methods in teleoperation. Unfortunately, creating reusable modules for robotic systems is a challenging task due to widely varying demands and designs; even small changes in e.g. the kinematic structure can demand for extensive modifications to a design. Moreover, certain task specific and specialist applications demand for dedicated control architectures with strictly defined specifications.

However, since all bilateral teleoperation systems share some typical characteristics and problems, there are possibilities for generalizing certain *robot independent*, but *control specific* parts. Research regarding bilateral control has focused primarily at:

- Overcoming issues caused by communication delays. The main objective is to always maintain a stable system in the presence of delays.
- Achieving an as high as possible degree of transparency by providing a good sensation of telepresence to the (human) operator, even in cases of significant (varying) delays.
- Guaranteeing stable and intuitive interaction by providing accurate force reflection and qualitative contact stiffness.
- Aiming for high tracking accuracy and low drift by minimizing transient and steady-state errors.

Since the issues mentioned are inherent to all (bilateral) teleoperation systems, the benefits of creating reusable control elements for such systems are numerous. Several researches have aimed at creating modular and distributed controller designs in order to provide standardized methods of control. However, most of these strategies are either too specific to address a wide variety of systems or provide only limited functionality.

1.3 Goal of the Project

Given the fact that modular controller design is complex and intrinsically limited in some aspects, the main focus will be on generalizing signal interfacing and combining this with some well known strategies for achieving acceptable system performance. The work presented intends to contribute to a generalized, robust control architecture for bilateral teleoperation systems and provide certain reusable modules for ‘off-the-shelf’ implementation. This should result in a sort of *controller library*, enabling a designer to choose basic functional elements that can cope with the typical issues related to teleoperation. Since creating a completely modular, reusable controller design is not possible due to the inherent differences particular to a specific design, a modular architecture that differentiates between robot specific and behavioral specific control elements is pursued. The robot specific elements refer to those parts that are unique to each robot design (e.g. kinematics and joint/actuator control) while the behavioral specific elements impose behavior depending on a generalized task or coordinate frame (e.g. Cartesian control, interaction control & trajectory tracking).

An appropriate generalized interface strategy should be applied throughout the model to guarantee easy insertion of modules. Additionally, several standard control modules will be developed for providing basic functionality. Additional control methods for improving performance are reviewed and possibilities for integrating these methods into the proposed architecture are considered. Design of the controllers is done using 20-Sim and bond graph modeling. In order to verify the design and control modules, simulations will be performed on a model of the KUKA youBot. For this robot, the University of Twente has participated with the BRICS project, which’s prime objective is to structure and formalize the robot development process. It also provides tools, models, and functional libraries, which help accelerate this process. A bilateral teleoperation system for this robot is present and will serve as the basis for validating and verifying the results of this research.

The main objective of this project can be generalized as follows:

“Design of a reusable teleoperation control architecture based on existing control methods, for the youBot”

In order to be able to complete this objective the project has been subdivided into a number of goals:

- Evaluate existing control methods for teleoperation systems
- Design a generalized controller framework and interface strategy
- Modularize the controller architecture to provide reusable elements for similar applications
- Explore tuning of the control modules and the effect on overall system behavior
- Verify performance and reusability of the design by doing simulations with the youBot

1.4 Outline of the Report

The report is organized as follows: in *Chapter 2* the relevant background information on bilateral control of teleoperators is covered and typical issues encountered with these systems. The primary goals in bilateral teleoperation systems will be identified and a number of existing control methods for coping with particular problems will be addressed. Finally, an existing model for the KUKA youBot is presented and used as a test setup for implementation and verification of the complete design. In *Chapter 3* the analysis of the background information is presented and the main design objectives for obtaining a modular and reusable controller architecture are proposed. A generalized interface structure is presented and the reasons for using wave variables and impedance control are motivated. Finally a proposal on how to modularize the overall control architecture is put forth. In *Chapter 4* the low-level design of the control modules is treated. Also, considerations with regard to the implementation of certain control methods are outlined and possibilities of using different additional control with the proposed design will be evaluated. In *Chapter 5* the verification of the overall system and the results is covered. Simulations are performed to investigate the effect of tuning controller settings and to determine the effect on system behavior and task performance. Explorative simulations show how the overall system response is influenced and how this affects the quality of control and user-machine interaction. To conclude, in *chapter 6* the final concluding remarks and recommendations are presented.

2 Background

This chapter covers necessary background information for those readers who are not familiar with the subject.

2.1 Teleoperation

Mankind and his insatiable desire to explore the unexplored and discover the undiscovered has put himself in a wide variety of challenging situations and locations. A lot of these places are too dangerous to work in or do not support life altogether, even with protective measures, while other places are simple beyond reach. Nevertheless, there is a growing desire to investigate and operate in these regions, creating a vast demand for remotely controlled machines. Additionally, the development of the internet has quite literally brought the world to our fingertips, allowing for almost instantaneous communication (to the human response) with virtually any place on earth. This has made it possible for specialists to operate their expertise at remote locations without having to travel there first. Needless to say, this has generated an enormous demand for *teleoperation* systems and the desire to optimize these systems and their development processes.

2.1.1 Introduction

As was briefly discussed in chapter 1, there are two fundamentally different ways of communication within a teleoperation system; the one being *unilateral* (one directional; commanding) and the other *bilateral* (two directional; sensing and commanding). Additionally, some research is dedicated to *trilateral* telemanipulation, referring to two-master / one slave configurations. Examples of the latter include (Malysz and Sirouspour, 2011), where a setup is considered in which control of a robot is distributed among two master controllers. The primary master controls the telemanipulator motion while a secondary master is used for controlling interaction with the environment. Alternatively, in (Li et al., 2013) an application is presented where a first master controls a primary task frame and another master manipulates a secondary task frame (e.g. separately controlling a robot arm and its end-effector).

Teleoperation systems typically aim to provide an operator with a means to remotely carry out a complex task requiring direct human control. By adding force feedback a bilateral teleoperation system is established, which can considerably improve an operator's ability to perform complex tasks, see (Niemeyer and Slotine, 1991). On the local side a human operator interacts with a master device (e.g. a *joystick* or *haptic device*) and on the remote site the slave device (the *remote robot*) is interacting with an environment. Connected by a communication medium with possibly non-negligible time delay, the master and slave systems exchange information. Figure 2.1 shows a schematic diagram of a typical bilateral telemanipulation system. In this diagram, $\dot{x}_m$ and $\dot{x}_s$ are master and slave velocities, F_h is operator torque, F_e is environment torque, F_m is the feedback force from master controller to master manipulator and F_s is the reflected force from the slave manipulator to the slave controller. For the sake of generality the direction of the signals has been omitted as well as the type of signals being exchanged over the communication channel.

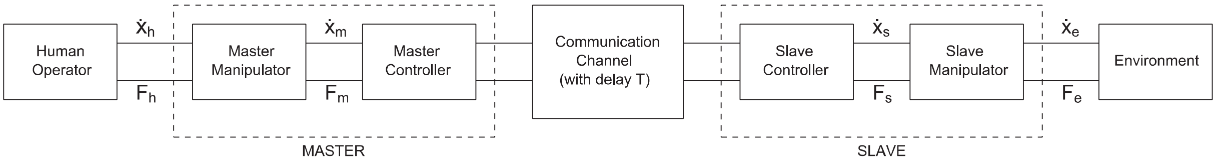


Figure 2.1: Typical representation of a bilateral telemanipulation system.

The most important goals for a teleoperation system are; to ensure a stable connection between the human operator and the remote robot and to provide faithful and intuitive feedback to the user. The latter is accomplished by transmitting the contact forces of the remote robot back to the operator, which is referred to as *force reflection*. Unfortunately, in the presence of transmission delays, force feedback can have a strong destabilizing effect on the system. This instability in particular can result in overall system failure and can cause dangerous situations, though a lack of accuracy can just as easily lead to undesired behavior (for example in telesurgery, micromanipulation and other precision applications). Also, due to delays in the communication medium, an operator will experience a sense of distance from the workplace, which can drastically complicate an operators' task. Besides stability, providing the user with a good sense of 'being at the remote site' (referred to as *telepresence* or the *transparency* of the system) is another important topic in teleoperation, and currently one of the most pursued objectives in research regarding teleoperation.

2.1.2 Key issues in telemanipulation

Time delays in the communications can be regarded as the primary concern in teleoperation systems (Cui et al., 2003). The presence of a delay in whatever form always adversely affects system behavior, degrading the overall system performance. Additionally, in order to avoid instability and poor performance when designing control for telemanipulation purposes, *interaction* with (unknown) environments is another important concern (Secchi et al., 2001). Due to contact dynamics (in manipulation tasks), response uncertainties and force disturbances (due to e.g. collisions), there is always a serious risk of undesired system behavior. To that end, interaction, in both the human-robot and robot-environment case, is a significant topic to consider. Because the primary purpose of most teleoperation systems is to interact with a remote environment or manipulate remote objects, interaction issues are inherent to any bilaterally controlled teleoperation system consisting of separate interacting physical subsystems. The manifestation of both issues is a direct consequence of the closed loop behavior of a delayed force-reflecting design, and therefore a fundamental property of in fact any bilateral teleoperation system.

2.1.3 Primary goals

As noted in chapter 1, providing *stability* is considered the most important aspect in creating a well functioning bilateral teleoperation system. Clearly, the goal is to guarantee stability under any circumstance; both in the presence of delays and when interacting with an environment. Another result of the bidirectional information exchange (where an operator typically commands a position and senses a force in response) is that the user perceives the system as a physically interpretable, dynamical entity. Consequently, in (Secchi et al., 2008) it is noted that besides stability, *transparency* is another major concern in the design of a telemanipulation system. This performance measure defines how well the system is able to convey to the user the perception of direct interaction with the remote environment, the goal being to approach the ideal situation (i.e. direct manipulation) as close as possible while maintaining stability. One final important notion widely addressed in literature, though not typically referred to as such, is the *accuracy* of the system, in (Aziminejad et al., 2006) referred to as the *tracking accuracy*. This performance related property is governed by two main characteristics; drift of the slave and tracking between master and slave.

One way to evaluate performance is by defining what a perfect teleoperation system should look like in terms of desired dynamical behavior. To this end, a textual definition is given in (Cui et al., 2003): "The system should behave as a zero mass, infinitely rigid, a tool (telerobot) with no viscous damping, and a teleoperator that has perfect force reflection, i.e. perfect transfer of momentum from slave to master on contact". Another definition from a control point of view is given in (Arcara and Melchiorri, 2001) where it is stated that an ideal telemanipulator should be stable for any delay T, present as low as possible inertia ($\cong 0$), display the same stiffness at master and slave side, and achieve zero tracking and no drift. Clearly such idealized systems do not exist; typically the master will have much lower impedance than the slave, a human operator's muscle stiffness is low, a human hand has little mass compared to large industrial manipulators and, due to the physical separation of master and slave, time delay is an inevitable factor. Consequently, a lot of research focuses on several specific strategies for improving system behavior, e.g. by minimizing the error between master and slave position, using predictive control strategies, increasing slave autonomy, or by providing an accurate model representation of the slave side.

2.1.4 Control solutions

In (Arcara and Melchiorri, 2001) a variety of control schemes for bilateral teleoperation with time delays are compared based on a set of criteria. The five aspects considered for comparison are; *stability*, *inertia & damping* perceived at master side, *stiffness* perceived by operator in case of interaction, *tracking*, and *drift.* These criteria roughly correspond to the goals presented in 2.1.3, noting that the dynamics of the system are highly determinative with respect to the user experience (i.e. the transparency of the system). It was observed that only passivity based schemes can guarantee intrinsic stability, both in case of time delays and interaction. With regard to interaction of the master and slave manipulators with the environment, several studies have compared and proposed different strategies. In (Stanczyk and Buss, 2005) two stiffness control algorithms and a position based impedance control scheme are compared. The impedance control strategy proved to have superior behavior, stable contact was achieved, and a successful assembly task was performed. Additionally, in (Chiaverini et al., 1999) a comparative study is presented, covering both static and dynamic model-based compensation schemes, the latter ones showing superior behavior. Good results were again obtained with impedance based control, though a hybrid impedance-force based control strategy yielded even better results. Sections 2.3, 2.4 and 2.5 will elaborate on control solutions with respect to the issues outlined in 2.1.2.

2.2 A generalized framework

When aiming for a reusable control architecture, a system description in terms of generalized control modules and standardized *interfaces* can provide a high degree of modularity. In order to allow reusability of the control modules, a framework should be adopted that allows modules to always be connected to other modules without having to redesign or map input/output variables to another domain. This means that apart from matching interfaces, the modules themselves should also represent domain independent concepts. Some widely used (mathematical) frameworks will be treated hereafter.

2.2.1 Passivity

Although passivity is an intrinsic feature of in fact any *physical* system (by the law of conservation of energy) this property does not necessarily hold for a *virtual* (i.e. non-physical) system. Because a virtual design need not be bounded, as it is evidently possible to create a fictitious system not obeying the laws of physics, there is always a chance of *virtual energy generation*. Specifically this generation of energy can result in non-passive behavior, causing overall system *instability*. As a solution to this issue, a well known and useful tool for guaranteeing *stability* of a system is by demanding its *passivity*. Moreover, any combination of passive subsystems will result in an overall system that is again passive and hence stable; see chapter 2 of (Alise, 2007). This strategy is widely adopted in control engineering and consequently also bilateral telemanipulation, as it offers a particularly robust approach for developing stable systems (Anderson and Spong, 1989a). The passivity formalism represents a mathematical description based on the intuitive physical concepts of power and energy, and can be applied even with nonlinear and time-varying systems. In practical terms, a system is said to be passive if it does not generate energy; i.e. energy can only be stored or dissipated (i.e. converted into heat) and thus, the energy supplied by any element in the system is limited to its initial stored energy. Mathematically, for a (sub)-system to satisfy the passivity condition, the following relation should hold:

$$\int \left(\mathrm{e}_\mathrm{n}^\mathrm{T} \mathrm{f}_\mathrm{n} \right) \mathrm{dt} \; < \; \int \left(\mathrm{e}_\mathrm{p}^\mathrm{T} \mathrm{f}_\mathrm{p} \right) \mathrm{dt} \; + \gamma^2 \tag{2.1}$$

which states that the net energy exiting the system ($E_{output} - E_{input}$) cannot exceed the initial stored energy. In this equation $\mathrm{e}_\mathrm{p}^\mathrm{T}\mathrm{f}_\mathrm{p}$ is the input-power, $\mathrm{e}_\mathrm{n}^\mathrm{T}\mathrm{f}_\mathrm{n}$ is the output power and γ^2 represents the energy initially stored in the system. It can be shown that without external inputs a passive system always converges to a stable state. In order to be able to guarantee stability of a controlled telemanipulation system, a (controller) design combining solely passive elements can be used to construct a system that is guaranteed to be stable. Additionally, passivity theory offers an excellent framework to passively design and tune the closed-loop dynamics, such that a desired stable behavior can be obtained.

2.2.2 Network theory and bond graph modeling

In order to describe a system from an energetic point of view, network theory offers a technique to create model subsystems based on one-ports and two-ports, by which energy may be exchanged with other systems or with the environment, see Figure 2.2. In the figure, f_p and f_n represent generalized flows, and e_p and e_n generalized efforts, the product of both being the power exchanged at the port. Subsystems interact with each other through these so called '*power-ports*', which describe an interface based on two conjugated variables whose product represents power. Note that these port variables need not necessarily be scalars but can also represent vectors. An approach which generalizes the concepts of power and energy interaction between any types of physical systems is bond graph theory. This modeling strategy allows for a power continuous description and analysis of dynamical system behavior. In bond graph modeling, so called *bonds* are used for connecting elements, representing the bi-directional exchange of energy by transporting efforts and flows in opposite directions along the connection. Information on bond graph modeling is readily available in literature, see e.g. (Paynter, 1961), (Breedveld, 1984), (Breedveld, 1985) and (Broenink, 1999).

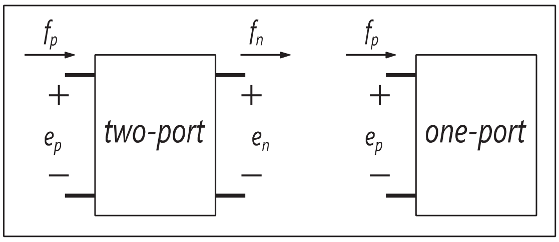


Figure 2.2: Schematic representation of a one-port and a two-port.

2.2.3 Screw theory

With regard to teleoperation, the related interaction between operator and robot is typically based on energy exchange in the mechanical domain. Moreover, because telemanipulation deals with robot control in the real world and hence operates in a 6-dimensional space (i.e. 3 rotations and 3 translations), a useful tool for describing motions and interactions arising in kinematics and dynamics of rigid bodies is *screw theory*. Using this mathematical framework, velocities and forces can be represented geometrically as twists and wrenches. This general approach allows for a mechanical system to be described and dynamically analyzed in a coordinate free way, and offers an intuitive way for designing controllers to shape the dynamic behavior at the systems ports of interaction, see (Stramigioli and Bruyninckx, 2001) and (Stramigioli, 1998). A short notational review can be found in Appendix A. Note that throughout this work the analogous concepts of force, wrench and effort, as well as velocity, flow and twist, can be used to describe the same variables.

2.3 Delayed teleoperation

As was noted in section 2.1.2, the general consensus in literature is that communication line delays are the main issue in teleoperation systems. Certainly in case of force feedback systems, where both sides depend on information from each other, the control loop can easily go unstable in the presence of delays (Niemeyer and Slotine, 2004). In fact, since transmission delays are by definition inherent to any system communicating over a distance, it is the primary concern to address with regard to any successful telemanipulation controller design. As was treated in section 2.2.1, passivity theory can be used to show that any passive system without external inputs is stable. By representing a teleoperation system as a series of one- and two-ports and by guaranteeing passivity of all these subsystems, the complete system can be proven stable. With regard to Figure 2.1 this means that all individual blocks should be passive. This is exactly where the instability issue associated with teleoperation originates from; delays in the transmission line cause the communication block to behave as a non-passive element. Consequently, the solution is to provide a method by which the communication channel can be rendered passive again, which is precisely what can be achieved with scattering theory.

2.3.1 Scattering theory and wave variables

In order to obtain a passive and thus stable communication channel, an elegant approach has been proposed in (Anderson and Spong, 1989b). Using passivity and scattering theory, a control law for bilateral teleoperators is presented which guarantees stability independent of time delays occurring in the communication medium. By mapping the conjugated power variables force and velocity (or more generally effort and flow) to wave variables, the communication block can be made to behave as a lossless element. The communication line in that case only temporarily stores the information during the length of the delay, and is therefore passive. The wave reflection problem arising with this approach has been solved in (Niemeyer and Slotine, 1991) for a single degree-of-freedom (DOF) linear, time-invariant (LTI) teleoperator system. The transformation is defined as:

$$u_m = \frac{1}{\sqrt{2b}}(F_m + b\dot{x}_m) \qquad u_s = \frac{1}{\sqrt{2b}}(F_s - b\dot{x}_s)$$

$$v_m = \frac{1}{\sqrt{2b}}(F_m - b\dot{x}_m) \qquad v_s = \frac{1}{\sqrt{2b}}(F_s + b\dot{x}_s) \tag{2.2}$$

where $\dot{x}_i$ are the velocities, F_i the forces, and u_i and v_i the forward and return waves, with $i = m, s$ the master and slave respectively. This wave transformation is bijective, meaning that the equations are one-to-one and invertible, with only the forward and return waves fixed. Rewriting the equations yields expressions for decoding the wave variables back to power variables. These equations are governed by a single scaling parameter 'b' called the *wave impedance*, which characterizes the impedance of the channel. This parameter can be tuned arbitrarily with the only constraint being that it must be strictly positive. This characteristic impedance directly influences the system behavior and as such should be tuned in order to achieve an acceptable response. However, because the impedance parameter is a scalar quantity and the only tunable parameter available in the 1-DOF wave equations, choosing a correct value is a straightforward exercise. Relating the wave impedance to additional control laws in the one dimensional case has been addressed extensively in (Niemeyer and Slotine, 2004). With the wave transformation, in any case u_m serves as the output variable of the transformation on the wave side (entering the communication line) and v_m as the input variable (coming from the line). For the power side the choice of input variable is free, and an expression can be easily obtained by rewriting the wave equations.

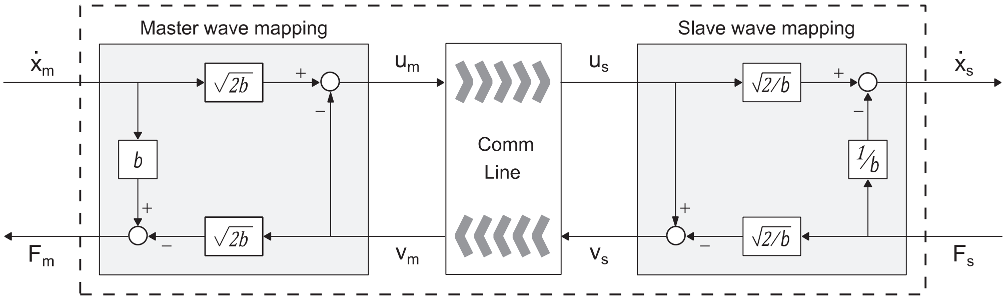


Figure 2.3: Schematic drawing for wave variable mappings.

The schematic representation for a wave variable mapping, based on Equations (2.2), can be found in Figure 2.3, where $\dot{x}_i$ and F_i again represent velocity and force, and u_i and v_i represent the forward and return waves respectively. The equations for this specific situation, where either side of the transformation supplies one input value and receives an output value, can be written as follows:

$$u_m = \sqrt{2b}\,\dot{x}_m - v_m \qquad \text{and} \qquad F_m = b\dot{x}_m - \sqrt{2b}\,v_m$$

for the transformation port at the master side, and **(2.3)**

$$\dot{x}_s = \sqrt{\frac{2}{b}}\,u_s - \frac{1}{b}F_s \qquad \text{and} \qquad v_s = u_s - \sqrt{\frac{2}{b}}F_s$$

for the slave side. For the total power entering the system (dotted line) depicted in Figure 2.3 this means that:

$$P_{in}(t) = \dot{x}_m(t)\,F_m(t) - \dot{x}_s(t)\,F_s(t)$$

With respect to the passivity demand, for the energy balance of this system it must then hold that:

$$E_{stored}(t) < E_{in}(t) + E_{stored}(0)$$

where $E_{stored}(t)$ is the energy stored in the system at time t and $E_{stored}(0)$ the initially stored energy. $E_{in}(t)$ represents the energy flowing into the system, which can be written as:

$$E_{in}(t) = \int_0^t P_{in}(\tau)\ d\tau = \int_0^t \big(\dot{x}_m(\tau)\,F_m(\tau) - \dot{x}_s(\tau)\,F_s(\tau)\big)\ d\tau \tag{2.4}$$

Assuming a constant time delay T, the wave outputs at the communication block can be written:

$$u_s(t) = u_m(t-T)$$
$$v_m(t) = v_s(t-T)$$

Filling these variables into Equation (2.4), then expanding and rewriting, yields the following result (for a more extensive derivation the reader may refer to appendix A):

$$\begin{aligned} E_{in}(t) &= \frac{1}{2}\int_0^t \big({u_m}^T(\tau)\,u_m(\tau) - {v_m}^T(\tau)\,v_m(\tau) + {v_s}^T(\tau)\,v_s(\tau) - {u_s}^T(\tau)\,u_s(\tau)\big)\ d\tau \\ &= \frac{1}{2}\int_{t-T}^t \big({u_m}^T(\tau)\,u_m(\tau) + {v_s}^T(\tau)\,v_s(\tau)\big) \geq 0 \end{aligned} \tag{2.5}$$

which confirms that with constant delays the communication channel is indeed passive with the wave variable mapping. Though this is a solid result, drawbacks of the wave variable method include an increase in overshoot and a decrease in manipulation speed with increasing time delay. Certainly with larger delays, performance decays rapidly. Besides the stability issue, delays can also greatly influence the transparency of the system.

2.3.2 The multi-DOF case

As an extension to the wave variable transformation, more general approaches have been suggested in order to be able to address systems with more than one degree of freedom (DOF). For this multi-DOF case, the wave transformation parameters relating the power and wave variables are matrices instead of a scalar quantity. With reference to the wave equations from (2.2), a more general form has been presented in (Munir and Book, 2001a). For the master side the *multi-DOF wave equations* can be written as:

$$
\begin{aligned}
\boldsymbol{u}_m(t) &= A_w \dot{\boldsymbol{x}}_m(t) + B_w \boldsymbol{F}_m(t) \\
\boldsymbol{v}_m(t) &= C_w \dot{\boldsymbol{x}}_m(t) - D_w \boldsymbol{F}_m(t)
\end{aligned}
\qquad \textbf{(2.6)}
$$

where A_w, B_w, C_w and D_w are the $n \times n$ *scaling matrices* (or alternatively *wave variable impedance matrices*) which relate the power variables to the wave variables and can be used to tune the communication line dynamics. Furthermore, $\boldsymbol{u}_m$ and $\boldsymbol{v}_m$ are the wave variables, and $\dot{\boldsymbol{x}}_m$ and $\boldsymbol{F}_m$ represent the velocity and force respectively. On the slave side similar relations hold for $\boldsymbol{u}_s(t)$ and $\boldsymbol{v}_s(t)$. Additionally, conditions for the scaling matrices can be determined in order to guarantee passivity, see (Munir and Book, 2001a) or (Munir and Book, 2001b). For the power into the communication channel it holds that (see appendix A):

$$
P_m = \dot{\boldsymbol{x}}_m^T \cdot \boldsymbol{F}_m = \frac{1}{2}\boldsymbol{u}_{\boldsymbol{m}}^T \boldsymbol{u}_{\boldsymbol{m}} - \frac{1}{2}\boldsymbol{v}_{\boldsymbol{m}}^T \boldsymbol{v}_{\boldsymbol{m}} \qquad \textbf{(2.7)}
$$

and similar for the slave side (refer to Equation (2.5)). Substituting Equations (2.6) into Equation (2.7) yields the requirements (Alise et al., 2009):

$$
\begin{aligned}
C_w^T C_w &= A_w^T A_w \\
D_w^T D_w &= B_w^T B_w
\end{aligned}
\qquad \textbf{(2.8)}
$$

and

$$
A_w^T B_w + C_w^T D_w = I \qquad \textbf{(2.9)}
$$

for guaranteeing passivity of the system. In (Munir and Book, 2001a) conditions are derived to ensure these requirements are satisfied. They specifically consider the case in which $C_w = A_w$ and $D_w = B_w$, with A_w a symmetric matrix, and reducing Equation (2.9) to $I = 2\, A_w^T\, B_w$ to prove system stability.

An alternative method for the multi-DOF case has been studied in (Stramigioli et al., 2002). Transforming power conjugated variables to *geometric scattering* variables allows for the interconnection of physical systems from an intrinsically geometric point of view. By adopting a port-controlled Hamiltonian description of the system, a multi-dimensional and coordinate free representation of the wave variable transformation is presented whereby *twists* and *wrenches* can be mapped to the wave domain. For the generic, multidimensional case the following equations describe the transformation, which can be rearranged to express either side of the mapping in one wave variable and one power variable:

$$
\begin{aligned}
\boldsymbol{s}^+ &= \frac{\boldsymbol{N}^{-1}}{\sqrt{2}}\,(\boldsymbol{e} + \boldsymbol{Z}\,\boldsymbol{f}) \\
\boldsymbol{s}^- &= \frac{\boldsymbol{N}^{-1}}{\sqrt{2}}\,(\boldsymbol{e} - \boldsymbol{Z}\,\boldsymbol{f})
\end{aligned}
\qquad \textbf{(2.10)}
$$

Here, $\boldsymbol{s}^+$ and $\boldsymbol{s}^-$ are the geometric scattering variables and $\boldsymbol{e}$ and $\boldsymbol{f}$ the power conjugated variables effort and flow respectively. In order to distinguish between scattering and 'regular' wave variables, for the geometric scattering variables, the notations $\boldsymbol{s}^+\ \boldsymbol{s}^-$ and $\boldsymbol{e}\ \boldsymbol{f}$ (as opposed to $u_i\ v_i$ and $\dot{x}_i\ F_i$) will be used throughout the report. Further, $\boldsymbol{Z}$ is the so called geometric *impedance matrix,* which is a symmetric positive semidefinite matrix, for which there always exists a symmetric matrix $\boldsymbol{N}$ such that $\boldsymbol{Z} = \boldsymbol{N}^{\boldsymbol{T}}\,\boldsymbol{N} = \boldsymbol{N}^{\boldsymbol{2}}$. Due to causality constraints either side of the mapping (from power domain to wave domain and vice versa) must supply one variable. For the wave domain side $\boldsymbol{s}^+$ always is an input variable but for the power domain side either of the two power variables can theoretically serve as an input, though there are multiple reasons for choosing to always compute $\boldsymbol{f}$ and $\boldsymbol{s}^-$ as a function of $\boldsymbol{e}$ and $\boldsymbol{s}^+$ (Stramigioli et al., 2000). Additionally, a clarification on impedance matching using Intrinsically Passive Controlled (IPC) physical systems is proposed for controlling the behavior of the telemanipulation system in a power consistent way. Section 2.3.3 will elaborate on this.

Though both the multi-DOF wave equations as well as the geometric scattering framework present a solution for guaranteeing passivity of the communication channel, neither approach addresses the complete family of passive scaling matrices. To this end, in (Alise et al., 2009) these works are further extended for the multiple-DOF case by deriving the whole class scaling matrices that satisfy the passivity condition. An extension to the class of passive scaling matrices as derived in (Munir and Book, 2001a) is presented, which obey the passivity conditions given in (2.8) and (2.9). It was shown that the complete family of scaling matrices can be derived based on certain requirements, and the effect of tuning the decomposed matrices was determined. This yielded following necessary and sufficient conditions to guarantee passivity:

1) A_w must be nonsingular (i.e. invertible).
2) $B_w = (1/2)\,(I + S_w)\,A_w^{-T}$ where S_w is any $n \times n$ skew-symmetric matrix.
3) $C_w = Q\,A_w$ where Q is any $n \times n$ orthogonal matrix.
4) $D_w = (1/2)\,Q\,(I - S_w)\,A_w^{-T}$

(2.11)

These equations are completely defined by providing three matrices; a nonsingular matrix A_w (i.e. $A_w\,A_w^{-1} = I$ and $det(A_w) \neq 0$), a skew-symmetric matrix S_w (meaning $-S_w = S_w^T$ and $trace(S_w) = 0$) and an orthogonal matrix Q (for which it holds $Q^T = Q^{-1}$ and $det(Q) = \pm 1$). Moreover, in (Alise et al., 2009), experiments have been performed showing how tuning of these matrices affects the overall system performance, and the following was concluded:

1) No matter which Q is used, the overall system output will not change.
2) As A_w changes, there is a tradeoff between speed and accuracy. In experiments users performed worse when A_w had larger terms off the main diagonal, reflecting to larger amounts of cross-coupling.
3) As the norm of S_w increases, the master and slave tend to experience more overshoot and longer settling time.

2.3.3 Wave reflections and impedance matching

In physical systems, waves are reflected at junctions and terminations where the impedance of the wave carrier (i.e. the transmission line) does not match that of the connected subsystem. Consequently, in case of teleoperation, wave reflections in the transmission scheme may occur at both the local and remote site if the impedances at both ends are not matched to the rest of the system. Although with the given scattering approach stability is guaranteed for any time delay, for increasing delays the amount of information in transit on the line at any given instant increases as well. Therefore, with increasing delays any impedance mismatch will result in the rebounding of increasingly larger amounts of information between the master and slave across the communication line. This can result in very large settling times for the teleoperator, and hence this energy received from the line should be absorbed by master and slave systems. As was recognized in (Niemeyer and Slotine, 1991), impedance matching has a great influence on system performance and as such appropriate scaling of the apparent impedance is a necessity.

As was noted in section 2.3.1, any expression for the wave equations will yield a system obeying the passivity principle, and hence produce a stable communication line. Intuitively it might seem obvious to let the port at the remote site have opposite causality to the port at the local site (see Figure 2.4); i.e. the power variable entering at the master side exits at slave side and vice versa. This way direct flow and effort paths are created between operator and telerobot, with the user commanding the velocity (indirectly the position) and the robot reflecting a force. However, both configurations yield a drastically different overall performance. It has been shown in (Niemeyer and Slotine, 1991) that there are multiple reasons for choosing the flow as an output variable; only in case of impedance causality can the system be matched to the transmission line impedance.

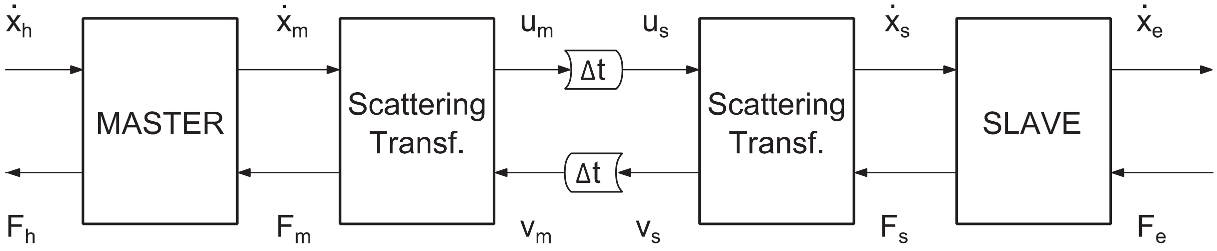


Figure 2.4: Master-Slave configuration with scattering transformation.

The issue of impedance matching has also been addressed for the geometrical case in (Stramigioli et al., 2000). It was deduced that impedance matching is independent of the state of the system and only depends on the system interconnection and the dissipative term present in the connected subsystem. Hence, energy-storing elements (such as compliances and masses) do not alter the interconnection or the dissipation and can thus be used without consequences on impedance matching.

With regard to information propagation throughout the model, in (Niemeyer and Slotine, 2004) it is noted that in a typical wave-based teleoperation design, one can distinguish three feedback paths; the immediate feedback in the form of damping, a path containing and based on wave reflections, and a third path that moves signals from the slave back to the master; see Figure 2.5. The first path does not travel through the delay and provides appropriate damping, in case of the geometric scattering approach characterized by the admittance $\boldsymbol{Z}^{-1}$. The second path causes a circular disturbance due to reflections which contain little useful information and should therefore be eliminated. The third path contains the important data and should be protected since it provides the force feedback to the operator.

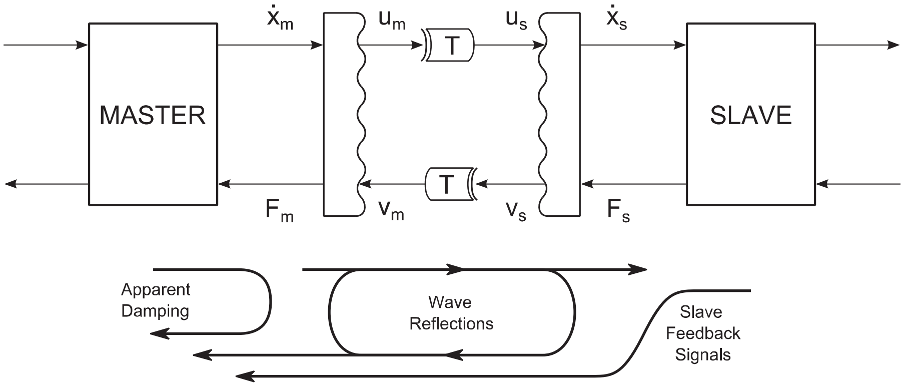


Figure 2.5: Feedback paths in a wave variable module *(based on* (Niemeyer and Slotine, 2004)*, Fig. 6)*

2.3.4 Varying delays

The original scattering formalism introduced in section 2.3.1 theoretically only preserves passivity for constant delays. However, a communication medium such as e.g. the internet, which is becoming more common in telemanipulation applications, is highly unpredictable (due to factors as congestion, bandwidth and distance). Such a varying delay modifies the equation from (2.5) and can be rewritten as follows (Lozano et al., 2002):

$$E(t) = \int_0^t P_{in}(\tau)\, d\tau = \frac{1}{2}\left\{\int_{t-T_1(t)}^{t} {u_m}^T(\tau)\, u_m(\tau)\, d\tau + \int_{t-T_2(t)}^{t} {v_s}^T(\tau)\, v_s(\tau)\, d\tau \right.$$

$$\left. - \int_0^{t-T_1(t)} \frac{T_1'(\sigma)}{1-T_1'(\sigma)} {u_m}^T(\sigma)\, u_m(\sigma)\, d\sigma - \int_0^{t-T_2(t)} \frac{T_2'(\sigma)}{1-T_1'(\sigma)} {v_s}^T(\tau)\, v_s(\tau)\, d\sigma \right\}$$

where $T_i'(\sigma) \coloneqq \frac{dT_i}{d\tau}$. The first two terms in this equation are positive definite, storing the wave energy during transmission. However, the last two terms may cause energy generation due the possibly negative terms inside the integral. For $(T_i' > 0)$ the delay increases and the communication channel produces energy, while for decreasing delays the channel absorbs energy and thus remains passive. Again, this production of energy may compromise the stability of the system. In order to overcome this issue a modification to the original architecture can be made by inserting a varying gain $f_i(t)$ after the time varying delay 'block' (i.e. on the receiving ends of the communication channel), see (Lozano et al., 2002). The approach can recover passivity of the channel provided a bound on the rate of change of the delay is known. This results in the following relation for the transmission line, where $T_i(t)$ represent the time varying delay:

$$u_s(t) = f_1(t)\, u_m\big(t - T_1(t)\big)$$

$$v_m(t) = f_2(t)\, v_s\big(t - T_2(t)\big)$$

This yields an energy equation for the communication channel where the last two terms become:

$$-\int_0^{t-T_1(t)} \frac{1-T_1'(\sigma)-f_1^2}{1-T_1'(\sigma)} u_m{}^T(\sigma)\, u_m(\sigma)\, d\sigma - \int_0^{t-T_2(t)} \frac{1-T_2'(\sigma)-f_2^2}{1-T_2'(\sigma)} v_s{}^T(\tau)\, v_s(\tau)\, d\sigma$$

These two terms can be cancelled by choosing $f_i^2 = 1 - dT_i/d\tau$. Alternatively, by choosing $f_i^2 \leq (1 - dT_i/d\tau)$ the communication line is passive for delays $dT_i/d\tau < 1$.

The disadvantage of this approach is that choosing a constant upper bound on the delay in a highly varying time delayed communication medium such as the Internet, might unnecessarily degrade overall performance. To that end, in (Rodríguez-Seda et al., 2006) different Internet-based teleoperation schemes are experimentally compared. These schemes were compared experimentally based on several criteria; stability, telepresence, tracking, complexity of design and the effect of delays and data loss. Based on experiments with four different internet models (defined in Table E.1), control schemes are rate based on several important criteria. Other solutions for coping with varying delays and optimizing tracking are shortly covered next.

2.3.5 Tracking and Drift

Though the scattering formalism offers a robust way for preserving passivity of the communication channel, the level of transparency that can be obtained has been criticized in e.g. (Franken et al., 2011) and (Lawn and Hannafrod, 1993). Moreover, because position information is only implicitly transmitted over the transmission line (in the form of a velocity), position drift and tracking errors due to non-idealities of the system are a common issue. To this end, several researches have focused on improving tracking performance by modifying the traditional wave variable scheme, such as e.g. (Benedetti et al., 2001) and (Ching and Book, 2005). A particularly elegant approach is by transmitting the wave integrals along with the regular wave variables (see (Niemeyer and Slotine, 1991)), by which the desired slave position is based on the master position, preventing any drift. Another solution for linking the slave position to the master is by communicating non-coded position information in addition to the waves across the communication line. In (Chopra et al., 2004) an architecture was developed using direct position control, which improved position tracking while maintaining comparable force tracking.

2.3.6 Wave filtering

In (Tanner and Niemeyer, 2004) it is noted that non-idealities may exist in physical systems which are typically not modeled. Furthermore, processing delays, structural resonances, and modeling mismatches may introduce additional dynamics into the system which are typically not taken into account. The consequence of this is that the system can become slightly active at higher frequencies which can lead to violation of the passivity principle. As was already recognized in (Niemeyer and Slotine, 1991) filtering of the wave signals can be done passively and therefore without effects on stability. Using a simple filter in the wave domain provides a means to effectively reduce the effect of reflections and smoothen the motion of the telerobot. Another beneficial consequence of filtering signals directly in the wave domain is that it provides a passive and systematic way to reduce noise.

Due to the wave variable mapping, additional dynamics (such as mentioned above) may be introduced into the system and their effects can accumulate, specifically in the wave domain itself; i.e. on the transmission line. The aforementioned wave reflections are an example of such introduced dynamics, though the impedance matching strategy introduced in 2.3.3 can eliminate these reflections. Nevertheless, it may still occur that disturbances exist on the line, possibly causing oscillations due to the internal (delayed) feedback loop. For instance, as discussed in (Tanner and Niemeyer, 2004), certain modeling assumptions can disrupt system performance. Mechanisms are typically modeled using ideal elements which describe the dominant dynamics, which often yields a perfectly acceptable approximation of the actual physical system for behavioral purposes. Nevertheless, amplifier, actuator, and sensor dynamics are often ignored and these non-idealities can lead to induced higher frequency components. Though constantly improving developments reduce undesired residual dynamics, there will always be undesired dynamics introduced into the system. To this end, several researches have suggested employing wave filters for improving robustness of the design. For example; by implementing a low pass filter in the wave domain, the generated energy can be dissipated, therewith reassuring passivity. Though such a filter adds phase lag to the system, this is in fact no more than some additional delay, which in the wave domain behaves in a passive way. Hence, the added phase lag due to the addition of filters directly in the wave domain does not compromise stability of the system.

2.4 Interaction & manipulation

Mechanical interaction with an environment is a fundamentally important aspect of robotic teleoperation systems (in most cases the sole purpose of the design) which has been extensively researched in history. Though early *position* and *force* control approaches have provided moderately successful results it is well known that these methods can easily cause instability and can only provide limited stability under very restrictive conditions. Furthermore, neither of these two methods is suitable to properly control the transition between contact and non-contact: Pure position control offers no meaningful contact information and is highly sensitive to disturbances when dealing with unknown environments. Pure force control on the other hand is not meaningful in non-contact situations due to the fact that no forces can be exerted in free space. Even more challenging is manipulation of objects, including for example grasping and positioning, due to the complex nature of multi-body interaction and possibly unknown geometrical and dynamical properties of the object.

2.4.1 Impedance control

A more effective method for describing the dynamic interaction between a robot and its environment has been proposed in (Hogan, 1984). Because interaction is based on the exchange of mechanical work between a manipulator and its environment an alternative approach is to consider a control strategy based on a combination of *efforts* (forces) and *flows* (velocities). By recognizing that the physical world behaves as an *admittance*, Hogan proposed that in order to properly control the interaction dynamics, the controller should behave as an *impedance*. By describing the complete system as an equivalent physical system, the dynamical coupling between the interacting subsystems can be properly considered. Regardless of their implementation the controllers are passive by nature.

2.4.2 Intrinsically Passive Control

A promising interpretation of the impedance control approach is presented by passivity based strategies which consider the energy exchanges in system. A grasping strategy proposed in (Stramigioli, 1998) presents a practical application of such an impedance controller and its stable behavior was proved. Several studies demonstrated the usefulness of this controller for robotic manipulators, see e.g. (Wimböck et al., 2008) and (Differ, 2010). Referred to as intrinsically passive control (IPC), the strategy is algorithmically passive and highly intuitive as all the concepts used for the design are physically interpretable. Being composed of passive elements, the control strategy can ensure stable behavior in all situations. Consequently, the algorithm controls the dynamic behavior of a manipulator, ensuring smooth transition from contact to non-contact.

In (Stramigioli et al., 1999) an application of the IPC algorithm is presented composed of a virtual object, compliances and energy dissipating elements; the direct physical equivalences of a mass, spring and a damper respectively. Figure 2.6 shows a physical interpretation of the control strategy in one dimension. It should be noted that its performance is not related to its intrinsically passive nature but rather depends on the dynamic model of the system and the tuning of the controller parameters. Generally, the IPC itself consists of two (power) ports by which energy can be exchanged between the two interacting systems; a supervisor on the one side, supplying energy to the system in order to perform a desired task, and the system to be controlled on the opposite side. A geometrical framework for the implementation of the IPC concept is developed in (Stramigioli, 1998) and experimentally evaluated in (Stramigioli et al., 1999).

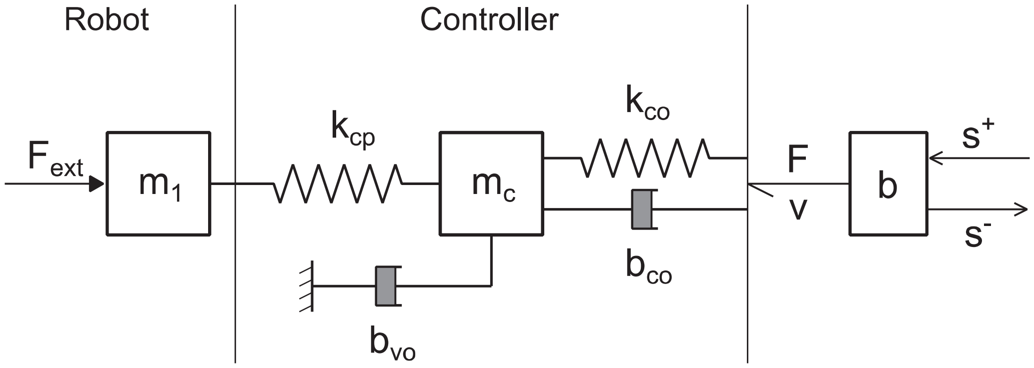


Figure 2.6: Structure of the IPC in a one-dimensional teleoperator using scattering transformation *(source:(Stramigioli et al., 2002), Fig.5)*

In the figure above, the elements that constitute the IPC controller are described by one dimensional masses, springs and dampers. In the geometric case these elements are represented by spatial compliances, virtual objects (inertias) and damping elements. In (Stramigioli et al., 1999) the equations for the elements used in the above IPC controller from are discussed from a geometric perspective. A *spatial compliance* is a generalized spring connecting two rigid bodies, and the dynamic behavior governed by the equation:

$$\begin{bmatrix} m_i^j \\ f_i^j \end{bmatrix} = \begin{bmatrix} K_o & K_c \\ K_t^T & K_t \end{bmatrix} \begin{bmatrix} \theta_i^j \\ p_i^j \end{bmatrix} \qquad \textbf{(2.12)}$$

where K_o is the *orientational stiffness*, K_t is the *translational stiffness* and K_c the *coupling stiffness*.
Equation (2.12) can also be written as (Stramigioli, 1998):

$$\widetilde{W}^i = \left[\ (m^i)^T \quad (f^i)^T \ \right]^T$$

with

$$\widetilde{m}_i^j = -2\, as\left(G_0 R_i^j\right) - \ as\left(G_t R_j^i \tilde{p}_i^j \tilde{p}_i^j R_i^j\right) - 2\, as\left(G_c \tilde{p}_i^j R_i^j\right)$$

$$\textbf{(2.13)}$$

$$\tilde{f}_i^j = -R_j^i \, as\left(G_t R_i^j\right) R_i^j - \ as\left(G_t R_j^i \tilde{p}_i^j R_i^j\right) - 2\, as\left(G_c R_i^j\right)$$

where G_o, G_t and G_c are the co-stiffness matrices which are defined by:

$$G_x = \tfrac{1}{2} tr(K_x)\, I^{3\times3} - K_x \qquad \text{with } x = o, t, c$$

The operation $as(\,)$ returns an *anti-symmetric* matrix by calculating $A = \frac{1}{2}(M - M^T)$, and the $tr(\,)$ operation represents the *trace* of a matrix and returns the sum of all elements on the diagonal.

For the *virtual object*, the dynamics are described by the general equations for a spherical rigid body:

$$P_b^b = P_b^b \Lambda\, T_b^{b,0} + W_{b\ tot}^{b} \qquad \textbf{(2.14)}$$

where $P_b^b = I_b^b\, T_b^{b,0}$ is the generalized momentum, $T_b^{b,0}$ is the twist of the body with respect to an inertial frame ψ_0, $W_{b\ tot}^{b}$ is the sum of all wrenches being externally applied to the object, and $P_b^b \Lambda$ is defined as:

$$\left(P_b^b \Lambda\right) := \begin{pmatrix} \tilde{P}_\omega^b & \tilde{P}_v^b \\ \tilde{P}_v^b & 0 \end{pmatrix}$$

Also, for the specific case of a spherical object $I_b^b = \begin{pmatrix} j\, I_3 & 0 \\ 0 & m\, I_3 \end{pmatrix}$, where I_3 is a 3×3 identity matrix.

Finally, for providing damping to the system, energy can be dissipated using a *spatial damper* with equation:

$$W_{b\ diss}^{b} = R\, T_b^{b,0} \qquad \textbf{(2.15)}$$

where $R \in \mathbb{R}^{6\times6}$ is a positive definite matrix representing a dissipation tensor in the coordinate frame ψ_b.

2.4.3 IPC in telemanipulation

Another application of the IPC concept that has been explored is its use in a (bilateral) time delayed telemanipulation system. Besides an exposition on geometric scattering, in (Stramigioli et al., 2002) a means to passively implement spatial telemanipulation is provided. By tuning the damping term that connects the controller to the communication line, wave reflections can be eliminated. Further research regarding the use of wave based communication in combination with IPC, address system behavior by taking into account the digital nature of the control (Secchi et al., 2003) and by presenting an evaluation method for transparency (Secchi et al., 2008). Several studies elaborating on this design framework focus on passivity and transparency in telemanipulation, (Franken et al., 2009), (Franken et al., 2011) and (Franken et al., 2011) among others.

2.4.4 Variable (spatial) springs

An extension on the IPC concept is addressed in (Secchi et al., 2001) with the introduction of the *variable length* and *variable stiffness* springs. This concept provides a power consistent way to alter the rest length and compliance of spatial stiffness without compromising intuition or passivity. By adding an extra power port to the spring the supervisory controller can modify these values and change the state of the whole element by introducing energy in it. subsequently commanding the robot to follow the motion of the virtual object. A schematic drawing of such a power consistent variable spring can be found in Figure 2.7. Though this depiction is a one dimensional version of such a spring, a coordinate free implementation can be found in (Stramigioli, 1998).

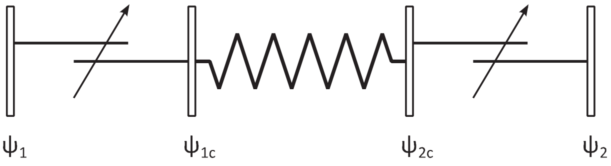


Figure 2.7: Representation of a variable length spring.

Expanding the spatial compliance with an additional port for varying the rest length of the spring can be done by modifying the twist relation:

$$T_i^{j,j} = T_b^{j,j} + Ad_{H_b^j} T_i^{b,b}$$

where the second term on the right side of the equation can be used to introduce an additional flow into the spring, hence passively changing its rest length.

2.5 Modular controller design

A lot of research has been dedicated towards creating modular robot designs for a wide variety of applications. The goal is always to provide a high degree of flexibility; designing control elements which are applicable to a wide variety of system. Some research focuses on modular software or controller designs, while others try to achieve hardware modularity (e.g. cooperative multi-robot designs, and robots consisting of reconfigurable modules).

A particularly interesting approach in order to achieve modularity of the controller design has been proposed in (Anderson, 1995). The SMART design (Sequential Modular Architecture for Robotics and Teleoperation) is a generic real-time control architecture for constructing telerobotic systems consisting of input devices, sensors and output devices. Based on two-port network theory, the architecture allows for rapid syntheses of complex telerobotic systems. Using passive one-ports and two-ports the telerobotic controller can be proved stable in spite of non-linear behavior of the operator, robot and environment. Behavior is represented using the effort-flow analogy, resulting in a number of basic element mappings, namely; inertia, damping, stiffness, Jacobian and velocity/force sources. In order to deal with stability issues, SMART considers both continuous and discrete system stability. By demanding all elements to be memoryless and all stored energy to be contained in linear-time-invariant stiffness and inertia elements, total system stability can be proved in the continuous case. For SMART it is a sufficient condition for stability to require that every single block obeys this passivity condition. Connecting these stable and passive single blocks together will then automatically provide for stability of the complete controller chain. However, due to discretization of the system, sampling delays can occur and result in algebraic loops which can easily cause instability. In order to cope with this problem all force and velocity signals at a module's port are mapped to wave variables. Finally, all module positions are propagated through the network in a feed-forward path for, among others, start-up synchronization.

Another approach based on software modularity can be found in (Scholl et al., 2001), being referred to as MCA (Modular Controller Architecture). By organizing the controller system in small standardized modules, reusability of these control modules in other project can be achieved, as well as a shortening of the training period for new users and a uniform software appearance. Control elements are represented in a hierarchical structure divided into *Modules* and *Groups*. Data handling between modules is done via edges which exchange sensor data and control data, flowing upward (from the sensors) and downward (to the actuators) respectively. Using this strategy stable operation of a variety of robots has been achieved. Though this research is based on creating dedicated pieces (i.e. classes) of software from an object-oriented programming point of view rather than a from a control perspective, the general structure offers an exemplary interface and framework that could be used in a reusable controller design.

2.6 The youBot

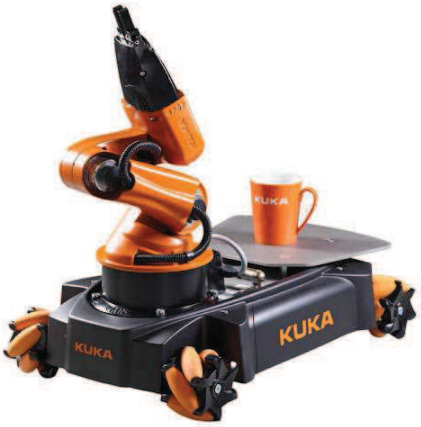


Figure 2.8: The youBot (*source: flickr.com/photos/kukayoubot)*

The youBot is a robot developed for educational purposes which consists of an omnidirectional platform, a five degree-of-freedom robot arm and a two-finger gripper (see Figure 2.8). It is an open platform which is operated and controlled from open source software aiming to serve as a reference platform for hardware and software development in mobile manipulation. For additional information regarding the youBot one may refer to (Locomotec, 2010) or (Locomotec, 2013). The University of Twente cooperates with the BRICS project in developments regarding the youBot, for which the development framework *BRIDE is available*, which stands for “BRICS IDE” (Integrated Development Environment). BRIDE is a model driven engineering tool chain based on Eclipse, developed as part of the BRICS project. It provides the basic framework for meta-model definition and transformation within the Eclipse Platform using the BRICS Component Model. Currently, two component frameworks are supported: ROS and Orocos, the latter provided with possibilities for real time operation. The Orocos Real-Time Toolkit (RTT) is based on a C++ framework targeting the implementation of control systems. For development of control modules and simulations, the 20-Sim modeling and simulation package can be used from which the necessary C++ code can be generated.

2.6.1 The youBot model

In order to be able to experiment and evaluate the design within this project, the KUKA youBot will serve as the main simulation platform, as was mentioned earlier. Though this report focuses on simulations, using the youBot will also allow for easy expansion and future experimenting on a real setup. In (Dresscher et al., 2012) a model for the youBot has been developed which will be used as the base model in the current work. The architecture of the arm model is modular, consisting of links and joints which can be interconnected to form any serial kinematic chain with certain specified dynamical properties. For the youBot platform an alternative dedicated model has been developed modeling a body with four mecanum wheels interacting with a floor surface. For a more detailed review of these models see appendix A.

2.6.2 Gravity compensation for the youBot

From the existing youBot model a gravity compensation controller can be easily deduced: By removing all dynamic elements from the joint and link modules, only retaining the kinematic information and the gravity sources in the links, the effect of gravity on the joints is calculated. Additionally, since the model is constructed using bond graph modeling the causality of the joint inputs should be reversed; the original youBot model demands a force input whereas the gravity compensation model should output the effect of gravity by means of a resulting force on each joint.

3 Analysis

In order to determine the aspects most important for the design of a reusable control architecture, a thorough analysis based on the background information (as presented in chapter 2) is essential. This chapter will discuss the requirements and criteria which need to be met in order to achieve the project goals, evaluate control frameworks and methods, and conclude with the design of a general architecture for teleoperation systems.

3.1 Introduction

As has been pointed out in chapter 1, the objective is to achieve *controller reusability* for bilateral teleoperation systems, by evaluating and extending on existing controller frameworks from a general point of view. Based on that, the methods selected for integration are those which meet the criteria and are most appropriate with regard to reusability and modularity. Additionally, any control method posing as little as possible limitations to the overall system organization (i.e. restricting users to modify or add to the design) has preference over methods that demand for strict design conditions. If needed, additional control elements are considered and integrated into the overall control architecture. The next sections will address following subjects:

- Evaluate requirements and criteria
- Determine approach and discuss design methods
- Select framework for control architecture

This results in a reusable controller framework which can then be evaluated using the youBot platform.

3.2 Requirements and criteria

This section examines the *feasibility* of a modular reusable controller architecture for teleoperation, and the *requirements* this entails with respect to the design. The requirements can be formulated as follows:

- The controller design should guarantee overall system *stability*
- The design should handle delayed tasks and interaction smoothly and intuitively; a measure for this is the *transparency* of the system
- Tracking between local and remote robots should be optimized and drift be minimized, i.e. an acceptable level of *accurac*y should be achieved
- The architecture should be *flexible* enough to provide an operator with the means to tune the system dynamics to a specific task
- The controller architecture should be *reusable* in an as-wide-as-possible range of teleoperators
- In order to provide for quick assembly of control elements, the design should be *modular*

Aiming for a design that is both robust and flexible, yet applicable even in dedicated systems, generally is a conflicting requirement. As a result, the goal should not be to design a control scheme which affects many parts of the system or comprises all important control elements in one design. Rather the goal is to provide a modular design consisting of control blocks with little but specific functionality which can be easily combined. Considerations on how to guarantee acceptable performance using existing control methods in a general design will be reviewed and extended if necessary, in order to acquire the desired behavior. The next sections will review the requirements and consider the implications for a modular reusable controller design.

3.2.1 Ideal telemanipulation

In order to select the control methods most suitable for a general telemanipulation controller design, the behavior for a perfect telemanipulation system can be used as the ideal goal. Based on the comments from section 2.1.3, an overview of the criteria and their respective desired goals can be established, see Table 3.1. Note that though these goals seem quite demanding, they only serve as a *guideline*.

CRITERIUM	GOAL
Stability	Always Stable
Inertia & Damping to Master	Inertia $M = 0$ & damping $B = 0$
Stiffness to Master	$K = \infty$
Slave tracking	No error: $\delta = 0$
Drift between Master-Slave	No drift: $\Delta = 0$

Table 3.1: Criteria for ideal telemanipulation

The goals seem clear; the system should always be stable, there should be *NO* drift, tracking should be perfect (i.e. *NO* positional error between master and slave), and the dynamics of the master-slave interconnection should comprise of *NO* inertia, *NO* damping and a rigid connection (i.e. *NO* compliance). Obviously such a system can never exist, which is why the aim can only be to approach these goals as close as possible. There is however one criterion that should be met in all cases if one is to provide an adequate modular reusable design, namely *stability* under all circumstances. Stable in this context refers to *intrinsically stable* rather than *possibly stable* (under certain conditions), meaning that the system should not become unstable under any given physical circumstance. The other criteria should be addressed carefully, by weighing the importance of each with respect to all other goals. The next section will briefly evaluate the importance of each criterion for the current opted design.

3.2.2 Evaluation of criteria

In order to determine which control schemes are best suited within the desired control architecture, the important aspects to consider are (with ■■■■■ being most important and ■ being least important):

- *Stability*. Stability should be guaranteed for an as wide range of situations possible. With regard to bilateral teleoperation this means creating robustness to all disturbances most likely to be encountered within such a system. Since instability renders a system completely incontrollable and hence useless in the sense that there is no way to achieve the desired task, stability will always be the *prime* concern. ■■■■■
- *Transparency*. Lack of transparency, or *obscurity*, is characterized by a sense 'mushy' interaction to an operator, which degrades his or her ability to adequately control the remote robot. Creating a good sense of telepresence is therefore of great importance, as it enhances the resemblance to the non-delayed situation (i.e. as if one were directly interacting with the remote environment). Apart from the fact that it creates a close connection between operator and slave manipulator it also provides for more intuitive control, making operation of the system accessible to a wide range of users. Though delays will inevitably degrade the transparency, an intelligent controller design can considerably improve an operators' experience by providing reliable tactile information, enabling a user to perform more complex tasks. Though a lack of transparency can complicate a task and thus require for more trained operators, consequences are far less critical than with stability. ■■■
- *Accuracy.* Depending on the application, a high level of accuracy may be required; e.g. in precision tasks such as telesurgery. The two main concerns here are slave drift and master-slave tracking. Drift, or *position offset*, can arise as a consequence of errors that accumulate due to e.g. discretization effects or initial condition mismatches. This can result in bad repeatability, demanding for a system reset, which can be quite undesirable when in the middle of a complex procedure. Tracking on the other hand, can have even more dramatic consequences. For example; when manipulating a remote surgical knife, overshoot of the slave device can cause unintended injuries to a patient. A slave device which closely follows the master limits the tracking error between both sides and ensures good trajectory following by the slave. ■■■■
- *Flexibility*. Though not mentioned specifically in the previous chapters, in order to provide a greater sense of intuitive control, it would be desirable to be able to tune the dynamical properties of the coupling between the human operator and the remote manipulator. Though in some aspects this closely relates to the transparency criterion, the dynamics between operator and remote manipulator also determine how the user interprets the telerobot at the slave site. Moreover, an operator may wish to shape the dynamic appearance of the slave regardless of what is typically regarded as ideal. There certainly is additional value to a slower, heavier response in case of operating a large machine in comparison to handling for example a thin, flexible needle. ■■

Next, an assessment of the available control methods with respect to these demands will be determinative for choosing the best option.

3.3 Design methods and approach

In this section, the extent to which a general teleoperation design can be modularized into reusable control blocks is investigated. In order to identify the bottlenecks when aiming for a generalized design, the typical architecture for a bilaterally controlled design is addressed, and it is determined in what way the controller design can be modularized. An overview of such a system from a control point of view is presented in Figure 3.1. Roughly, the controller design can be divided into two parts; an *application* specific part and a *behavior* specific part. The software part is divided into an application specific part and a behavior specific part. The *application specific* part refers to those control elements that perform certain tasks (mainly mappings) that are specific to every unique robot design. Joint control, actuator control and the interpretation of sensor data typically belong to this category since they completely depend on the robot design. The number of joints, the degrees of freedom (DOF's), the kinematic relations and the number of actuators and sensors, as well as their respective implementation, are all features inherent to a specific design. To the *behavior specific part* belong those controllers that impose certain operations or tasks not bounded to a specific coordinate frame but rather to the operational space of the physical world. These controllers impose commands or tasks in a geometric way (i.e. their validity does not depend on perspective) and as such can be transformed to a desired coordinate system by using a generalized mapping. Examples include trajectory planning and following, interaction control and Cartesian control. In the following, an elaboration considering the relevance of this notion with respect to reusability will be treated.

This research addresses the upper section of the graph presented in Figure 3.1 (labeled 'software'). This part relates to the actual control elements, more precisely the equations and the interfaces between these blocks, which generate actuator signals from the sensory information. It does not include the physical interface itself from computer to robot, nor does it include the sensors and/or actuators themselves.

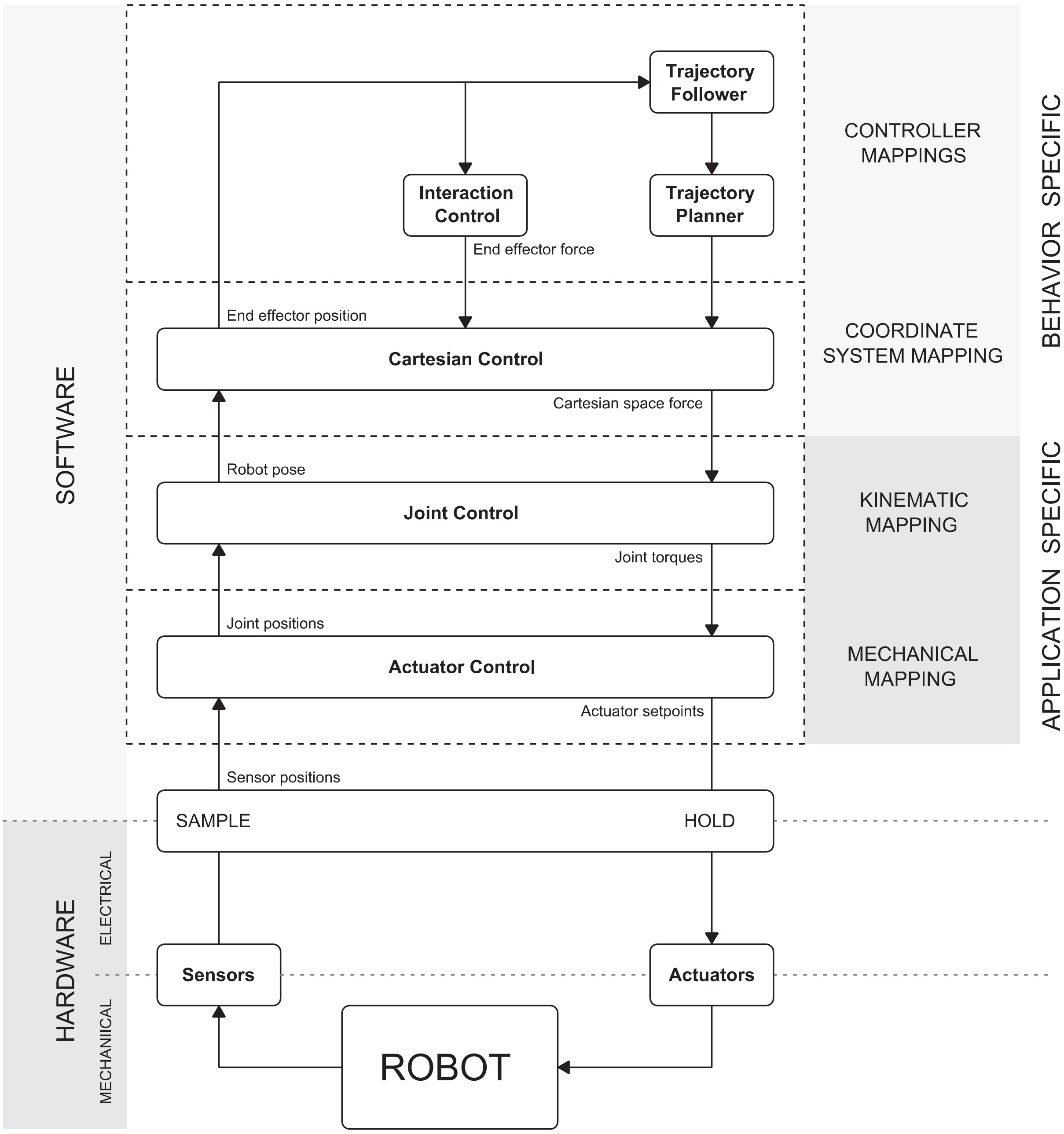


Figure 3.1: Control system overview *(inspired on (Brodskiy, 2014) Figure 2.13)*

3.3.1 Reusability

Reusability can be defined as the probability that a control element can be reused, to add functionality to new applications, with little or no modification to the element itself. In order to evaluate to which degree a reusable design is feasible at all, this section investigates the control structure for a typical teleoperation system. Since the application specific controllers need to be redesigned for every different robot, it means there is no straightforward way to define completely generalized interfaces for the control elements. Clearly, the number of inputs and outputs to a robot cannot be generalized, nor the coded information to and from the actuators and sensors. This means that certainly the application specific control modules need to be redesigned at least to a certain degree, for every different application. However, it *is* possible to provide functional blocks for modeling purposes, by creating modules that incorporate typical dynamic and kinematic properties for the mechanical elements of which robots consist. These standard building blocks can then be easily combined to describe a complete robots' behavior. An example of such a design strategy can be found in (Dresscher et al., 2012), where separate joint and link modules are used to construct a desired kinematic chain.

For the behavior specific modules though, it is possible to provide a greater level of reusability, since these modules operate in a desired virtual workspace. Signals from other spaces or domains can always be mapped to this space, without loss of information, and then be processed. Using a representation which is coordinate free and domain independent then allows for the unambiguous description of the controller without restricting oneself to a specific robot design. Typically, it is convenient to map signals from a specific robot, typically (angular) velocities and forces, to a coordinate frame such as *Cartesian space*. If then the behavior specific modules are chosen to operate in the same Cartesian space, these signals can be directly transmitted using a given interface strategy. With regard to reusability, the above exposition makes it clear that there are certain restrictions in generalizing the control elements and interfaces of a teleoperation control system. As such, the attainable level of modularity is restricted to incorporate modules which impose certain specific behavior in a generalized coordinate frame, and to define interfaces independent of the number of inputs and outputs.

3.3.2 Modularity

The degree to which the components of a system can be separated and recombined can be referred to as the modularity of the system. From the exposition on reusability presented in the previous section, it can then be deduced that reusability holds a strong connection with modularity. Clearly, there is a connection between the degree of modularity (the number, content, and size of modules) and reusability (the set of systems in which a module can be used). This conclusion can be substantiated by recognizing that in case of teleoperation the main reason for non-reusability is due to the varying *number* inputs and outputs (i.e. the number of sensors and actuators) and the *type* of control signals (e.g. velocity and force) used by the remote robot, since these are fixed by design. Clearly, an important aspect to consider is to which extent a design can and should be modular in the first place, and what degree of modularity is still intuitive to a developer. By breaking the control elements into increasingly smaller modules, possibly down to one-DOF signal paths, the architecture becomes increasingly modular, and also reusability of the modules is increased. However, a high degree of modularity typically yields a design which is highly fragmented, with modules containing little but very specific behavior. Though this adds to the reusability of the control modules, it decreases their functionality, which can considerably increase development time. Since the goal of this research is to *de*crease development time, it appears a careful trade-off between modularity and functionality is essential in achieving a satisfying result.

Due to the fact that the dimensions of robot specific parts are predetermined (a consequence of the specific design), completely generalizing all aspects of the interfaces for all control elements is highly improbable. One would have to generalize the number of DOF's and use the same variables throughout the model, satisfying the signal directions at all ports, which is far from obvious due to the varying number of sensors and actuators. Alternatively, a design framework based on separate signal paths for each controlled joint could be pursued, but this will significantly add to the complexity of the architecture, though decreasing the net computational complexity per element (see Figure 3.2.a). Also, a highly complex architecture generally yields a less intuitive design, while a high degree of computational complexity may require a large amount of computational power. Instead of generalizing the number of signals however, the type of signals being used can also be generalized for all control elements, without the task becoming too cumbersome. For the behavior specific part, complete generalization of the interfaces *is* an option, such that by placing the robot specific mappings close to the telerobot, reusability of the behavior specific modules can be maximized (Figure 3.2.b). These notions again emphasize the need for a careful consideration while designing the different control modules.

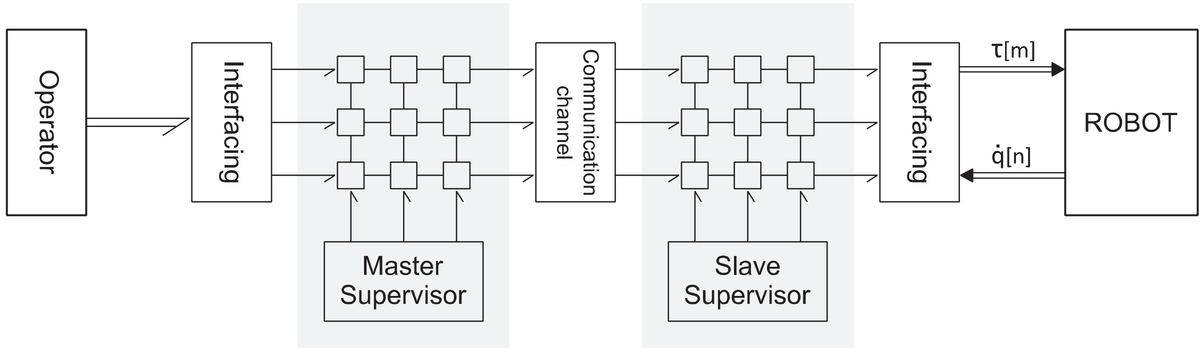
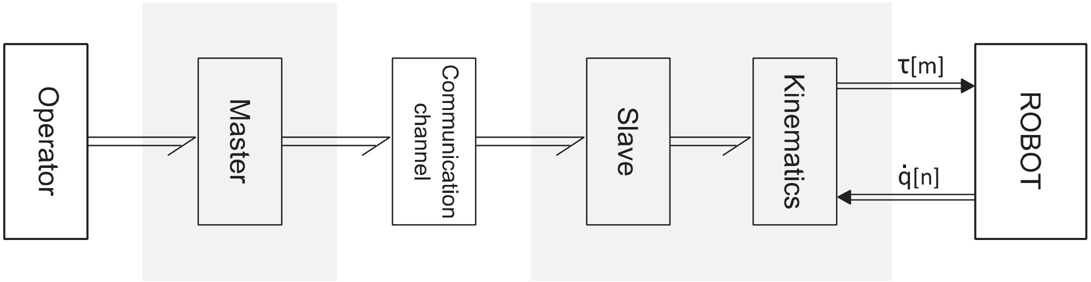


Figure 3.2.a: Maximizing modularity of the design yielding a highly fragmented architecture



b: Maximizing the behavior specific part of the control architecture

3.3.3 Passivity

As was put forth in section 2.1.3, the main goals of teleoperation from a control theoretic point of view are *stability* and *telepresence* (Hokayen and Spong, 2005). Because passivity-based control offers such a general solution with regard to preserving stability, and has been widely used throughout literature as an elegant way to guarantee stability, it will be the basis for the controller design in this research. To that end passive control methods will be employed to implement a design that is stable both in case of delayed communication and in case of interaction, the latter in particular for the transition from contact to non-contact and vice versa. Providing the human operator with a good sense of telepresence is another concern but in this case of secondary importance; the choice of control method is in this case restricted by the class of *passive* controllers. With regard to accuracy and shaping of the dynamic behavior a careful consideration should yield a design in which these criteria can be addressed in an 'open' manner, i.e. by supplying the user with enough tunable parameters to find the desired optimum for a given task. Conversely, the number of control parameters should not be so extensive that their functionality becomes obscured with respect to system response.

3.4 Controller framework

By now it is clear that the goal is to create a *passive* control architecture for teleoperation since in that case stability of the overall system can be guaranteed. From this one may conclude that the aspects most important to address are *communication delays* in the transmission line and *interaction* between separate subsystems, since these are the most significant stability compromising factors (recall section 2.1.2). Since delays are a fundamental consequence of communication over distance, any general controller design should incorporate a means to cope with these delays. Additionally, because the purpose of most teleoperation systems is to perform certain physical tasks at a remote location, providing the tools to establish stable interaction between the telemanipulator and the environment also will greatly add to the generality of the system. Both these matters have a comprehensive effect on the system and will therefore be construed as the most relevant areas to focus on. Two respective methods that comply with the passivity demand and have been successfully used, implemented, and even combined, are *wave variables* for delayed communication and *Intrinsically Passive Control* (IPC) for interaction. Additionally, control methods for improving tracking and preventing drift will be discussed, and an interface strategy for providing all necessary signals will be determined.

3.4.1 The wave variable method

Since communication relates to information exchange between two or more (sub-)systems, it typically does not belong to either of the systems taking part in the 'conversation'. Rather, the communication medium connects the interface ports of the communicating devices. Because the communication channel typically is a physical medium, it is not possible to directly control its characteristics, let alone modify them. The only controllable aspect is the actual information that the sender chooses to send over the line. Therefore, any influence one wishes to exert on the signal behavior and response of the channel should be performed prior to transmitting the data. This means that it is fair to conclude that the role of any delay compensating strategy can be primarily described as being an *interfacing strategy*. Since typical closed-loop position/force control methods have proven to be very prone to communication delays, a solution is to design an interface that maps these variables to delay insensitive signals, transmit these over the line and decode them back to the original variables.

In order to address the delay issue inherent to any teleoperation system, the wave variable method has been proven very useful, since it can guarantee stability under any time delay. Additionally, being based on the bidirectional transmission of velocities and forces (their product being power), the wave variable method also complies with the typical variables being used in robot control, namely position (integrated velocity) and force. Moreover, these variables directly relate to power flows and energy, which is highly relevant when aiming for a passive design. Section 2.3 has addressed the basic structure of this approach. A strategy for employing this method would be by equipping every control block with a wave variable based interface (as e.g. in Figure 3.3). However, this would lead to superfluous transformations within the system which could negatively affect overall system behavior and add unnecessary computations. A better approach would be to provide stand-alone interaction ports that map predefined power variables to wave variables and vice versa. Any two subsystems being connected to a communication line deemed prone to non negligible time delays can then be equipped with a wave variable interface.

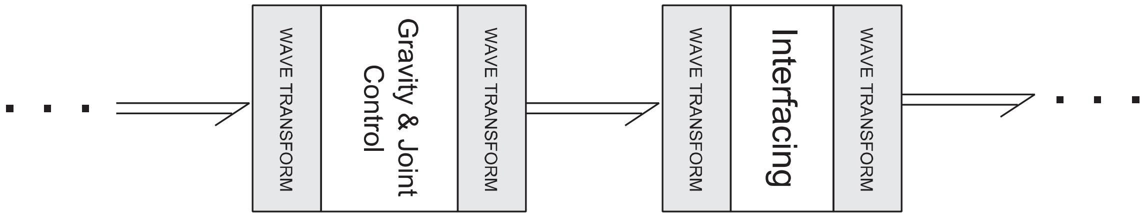


Figure 3.3: Wave transformation modules at each control element

3.4.2 Intrinsically Passive Control

Mechanical interaction of a manipulator with its environment or objects is such a fundamentally important aspect of robotic teleoperation systems, that a careful consideration of an appropriate control strategy is almost equally important as providing a proper controller for coping with delays. One particularly elegant approach used in (Stramigioli et al., 1999) is *intrinsically passive control* (IPC), which has been addressed in section 2.4. This control strategy emanates from the impedance approach for interacting physical elements and can therefore guarantee stability since it is passive in nature. Typically in robotic applications, an operator would like to control the position of the robots' end effector when aiming to direct it to a specific point in space. In that case position is the interesting control variable, be it the pose of the arm, the angles of the joints or the relative location of the tip with respect to some point in space. However, in case of actually interacting with the environment one could imagine that the relevant variable to control is the force. Certainly when grasping or manipulating a possibly fragile object it is important to be able to handle it with care. Because (bilateral) telemanipulation deals with both positioning and manipulation a suitable control strategy should be able to cope with both situations and provide a means to smoothly perform transitions from one to another.

In case of interaction the dynamic properties of the environment are important since they can describe the dynamical nature of an object. When attempting to control the manipulator, the physical interaction affects the controlled variable which introduces an error upon which the controller will act. The realization of this control loop clearly determines the quality of control but can even compromise system stability. Even though a system is stable on itself it can easily become unstable when being coupled to an environment. Such instabilities are referred to as contact instability and appear even in simple systems with simple controllers interacting with common environments. The IPC framework provides a means to adequately deal with the transition between non-contact and contact and the interaction itself while guaranteeing intrinsic stability.

3.4.3 Interfacing

Based on aforementioned control methods it is now possible to define an interface strategy for the modular design. The use of both wave variables and IPC for control motivates the use of power conjugated variables at the ports of interconnection, i.e. from an energy based point of view. In order to provide an as general as possible framework, the use of generalized efforts and flows is an appropriate way to communicate between subsystems. Since robotic systems are generally concerned with mechanical structures, their spatial properties and dynamical behavior, a geometrical description in terms of the generalized variables *twists* and *wrenches* is a straightforward choice. Moreover, using this interfacing strategy allows for easy connection to the modular modeling-approach that was presented in (Dresscher et al., 2012). For obvious reasons, i.e. the use of the youBot as test-setup within this research, the latter is a convenient incidental.

A modeling tool that is particularly suitable for communicating power conjugated variables is bond graph theory, since it is based on the bidirectional exchange of energy between components. This allows for a power consistent and domain independent description based on the use of physically interpretable elements. Accordingly, the use of bond graphs based on a geometrical interpretation will serve as generalized interface structure within this research. Though examples of such a framework exist, e.g. (Stramigioli et al., 2002), a reusable architecture in that specific sense has not yet been pursued. Also, tuning of parameters for both these control methods with regard to reusability has also not been extensively explored, as well as additional control with the purpose of serving an intended reusable modular architecture.

Finally, a good case can be made for using position information in addition to the power based interconnection, as this provides the means for synchronization and observer based control applications (Anderson, 1995). Moreover, transmitting position information from the master and/or slave controllers along the line allows for the application direct position control in case of varying delays. This motivates the use of position information propagation through the controller chain. An unambiguous way of describing the position of a rigid body in three dimensional space is by using homogeneous coordinates (i.e. twists and wrenches). This also again complies with the modeling approach used by Dresscher et al. (2012), and hence allows for straightforward connection to the existing youBot model.

3.4.4 Accuracy

As was noted in section 2.3.5, one of the challenges with wave variables is to achieve acceptable accuracy, especially in situations with varying delays. With large or varying delays, accuracy of the system deteriorates quickly resulting in poor tracking and coupling between master and slave systems. Therefore, if acceptable accuracy of the system is to be guaranteed even under such circumstances, the system will need additional means of control. Certainly in case of varying delays or missing data packets the wave variable design is highly prone to position mismatch due to the fact that no position information is transmitted. Hence, a lot of strategies aim to transmit direct position information across the line without compromising stability. To this end, in section 2.3.4 a reference was made to (Rodríguez-Seda et al., 2006), where several control schemes are reviewed. A table containing estimated values of the data from this study can be found in Appendix E.

The control schemes analyzed by Rodríguez-Seda et al. (2006) are the following:

(WS)	Wave scattering transformation
(DD)	Digital data reconstructing filter
(WI)	Wave integral and reconstruction filter
(P)	Proportional control
(PD)	Proportional derivative control
(WP)	Wave prediction with energy regulation
(AC)	Passivity based adaptive control

Table 3.2: Internet-based teleoperation control schemes

The values from Table E.3 can be used to determine an overall score per control scheme, summing the value of each scheme per criteria. This has been done in Table 3.3. In this table, all results have been normalized with respect to the best result obtained among all schemes, a unit value representing the best attainable score. For all error related criteria, a minimum value (1.00) yields the smallest error, which clearly is preferable. For the contact stiffness the result has been inverted, since a higher stiffness is preferable for transparency (recall Table 3.1). The last numerical value is complexity, which is a measure for the structural complexity and the computational complexity combined (both values have been multiplied and then again normalized). Also, the table documents whether the control method is wave based (i.e. using wave variables for transmission) and whether specific scheme guarantees passivity in case of varying delays (recall from section 2.3.4 that this is not a given). It should be noted that though the WS, P, PD, and AC schemes do not guarantee passivity by design, Rodríguez-Seda et al. (2006) noted that during experiments performed, all control techniques yielded a stable result. Unfortunately, it cannot be guaranteed whether this continues to hold for circumstances other than those used in the experiments. This can be considered a downside for these schemes since the goal is to provide stability for a large class of applications. From the comparison table it is clear that the total score for the WS and DD schemes is poor while the P, PD, and WP schemes rate average to good. The best results are obtained with the WI and AC schemes.

CONTROL SCHEME / CRITERIUM	WS	DD	WI	P	PD	WP	AC
Transient Error (free motion)	2.80	1.55	1.30	2.55	2.50	1.00	1.60
Steady-state Error (free motion)	5.11	3.67	1.00	1.44	1.78	1.00	1.56
Steady-state Error (constrained motion)	29.5	34.5	2.00	2.50	3.00	5.00	1.00
Contact stiffness	3.67	2.28	1.51	1.59	1.55	1.00	1.25
Force-reflection Error	1.47	2.01	1.00	1.69	2.06	1.69	2.04
Complexity	1.11	2.08	1.78	1.00	1.02	2.28	1.51
Wave based	Y	Y	Y	N	N	Y	Y
Guarantees passivity (for varying delays)	N	Y	Y	N	N	Y	N
TOTAL SCORE:	**43.66**	**46.09**	**8.59**	**10.78**	**11.91**	**11.97**	**8.95**

Table 3.3: Comparison of Internet-Based control schemes

Apart from preference for the WI method purely based on the total ratings in Table 3.3, there are clear reasons for selecting specifically this scheme. The WS and DD schemes are by far outperformed by the other schemes, showing larger transient and significantly larger steady-state errors. For the remaining schemes, the P, PD, and WP schemes also show larger errors than the remaining WI and AC schemes, be it less significant than with the WS and DD schemes. Since the goal of using either of these schemes is to compensate for the limited accuracy that can be achieved using wave variables, this definitely rules out the use of WS and DD schemes. With regard to contact stiffness and force-reflection error, the WI and WP schemes yield good results, while the P, PD, and AC schemes score slightly less. The least good results are again obtained using the WS and DD schemes. With regard to complexity it is worth noticing that the more complex schemes can guarantee passivity for time varying delays, while the lesser complex ones do not. If passivity is a demand, then in that case the WI-scheme is the best option with regard to complexity. Finally, the P and PD schemes are not wave based, which means that besides not being able to guarantee stability for time (varying) delays they also cannot be used with the opted design. Additionally, though the AC-scheme is wave based, it is also not possible to easily integrate the AC-scheme into the proposed architecture, as the scheme is based on different signals, which would require quite a design effort, since the IPC elements would have to be redesigned. To conclude; based on the above considerations the WI-scheme comes out as the best candidate, followed by the WP-scheme since this scheme is wave based and can guarantee passivity, the only reason for rejecting the scheme being due to a large constrained motion steady-state error. Using the IPC control strategy, position information is directly available from the virtual objects on both sides of the channel, hence the WI-scheme can be easily integrated.

Other possible approaches for guaranteeing tracking have been briefly noted in section 2.3.5 as well. One method employed in (Ching and Book, 2005) and (Niemeyer and Slotine, 2004) uses the master and slave positions to determine the drift error, and provide compensation by directly adjusting the wave command (*WC-scheme*). In comparison to the WI-scheme however, this would require to send position information across the line in addition to the wave variables. For applications with limited bandwidth this may compromise real time operation of the remote robot. If however, bandwidth is no restricting factor, one may even consider communicating position information from both master and slave controllers across the channel. An exemplary approach for such a design can be found in (Chopra et al., 2004) where corrective forces on both sides of the channel are directly calculated from the position difference between master and slave controller positions. For the remainder of this report this scheme will be referred to as the bilateral position exchange or “*BP” scheme*.

One of the main goals of the WI, WC and BP-schemes, is to provide absolute position feedback by observing the drift error between the two sides of the communication channel. Though these schemes offer a way to provide accurate tracking, the main difference between the methods is that the WI and WC schemes require additional control or filtering elements thus increasing structural complexity, and both the WC and BP schemes require additional transmission line bandwidth. An overview of the basic designs for these schemes can be found in Appendix F. Nevertheless, with regard to reducing tracking error, either of the schemes may serve as a suitable candidate in the proposed architecture. Because the IPC framework allows for a straightforward way to implement the BP-scheme (as will be explained in section 4.4.2) and transmission line bandwidth is not the restricting factor in a simulation exercise, it will be used here. It is worth noting though, that the WI-scheme is the only scheme which does not affect passivity of the scattering design since the position variables are encoded into wave variables themselves.

3.4.5 Wave Filtering

The concerns regarding disturbances due to non-idealities and other unwanted disturbances in the system, as discussed in (Tanner and Niemeyer, 2004), motivate the use of wave filtering since this could certainly add to the robustness and therefore generality of the design. Implementing a simple (first order) low-pass filter in the wave channel will eliminate high frequency noise, possible residual reflections or otherwise induced oscillatory behavior. As was outlined in section 2.3.5, filtering in the wave domain can be done without compromising stability of the overall system. Due to the fact that a filter induces phase shift, which is essentially nothing more than some additional delay, passivity of the channel will be preserved by the wave variable transformation. However, the added delay may have a negative effect on system performance and therefore, additional filtering should be used cautiously. Moreover, it must be assured that the roundtrip gain remains ≤ 1 since otherwise, just as with any other feedback controller design, the communication channel loop may still become unstable. Since this research does not aim at designing complex or dedicated filters, and there are plentiful examples readily available in literature, the use of wave filters in the current design will be restricted to first order low-pass filters.

In order to determine how to implement a wave filter, there are several considerations giving rise to a specific implementation. Clearly, it is important to dimension the filter with regard to a desired system bandwidth. Though this might seem a rather application specific choice, there are some characteristic features for every human controlled bilateral teleoperation system. By examining the bandwidth of the non-idealities of the system, a lower bound can be placed on the frequency at which problems arise (Tanner and Niemeyer, 2004). Additionally they state that “higher frequencies have shown to be very valuable feedback to the operator” implying that putting a too low bound on the bandwidth can possibly have a negative effect on the user experience. A solution to this is to only place a filter in the forward path, hence ensuring passivity but still allowing high frequency feedback. In addition, a more specific bound on the required bandwidth in the forward path is posed in (Aziminejad et al., 2006) where it is stated that “the human hand cannot generate trajectories with significant frequency content above 5-10 Hz”.

The above considerations put a clear limitation on the frequencies that have to be communicated from the master to slave. With this information it seems appropriate to incorporate a wave filter in forward path in the form of a first order low pass filter with a crossover frequency no lower than $10\ \mathrm{Hz}$. Nonetheless, a wave filter in the return path can be incorporated anyway to maintain symmetry of architecture. Placing the crossover frequency at an acceptably high frequency will preserve the valuable higher frequency force feedback. It should be noted that incorporating a wave filter in the forward (and possibly return) path may interfere with control methods such as the in section 3.4.3 suggested *wave integral scheme*, possibly requiring an alternative scheme to be used for improving tracking. Certainly when being dimensioned at $10\ \mathrm{Hz}$, a strict limit is put on the attainable system bandwidth. An approach to circumvent such issues would be to place a wave filter inside the actual scattering transformation block, more precisely in the direct path from $\boldsymbol{s}^+$ to $\boldsymbol{s}^-$ (see Figure 3.4) in case of using the geometric scattering approach. This way, the effort and flow variable paths remain unaltered while effectively filtering the inner feedback loop in the wave domain (with reference to Figure 2.5). Additionally, the wave integral scheme can also be implemented in this way. Hence, reliable signal exchange between master and slave systems is guaranteed while at the same time filtering the wave command. One may wonder whether this way of filtering causes issues due to bandwidth differences between the return wave $\boldsymbol{s}^+$ and effort $\boldsymbol{e}$. To that end, simulations will be performed so to evaluate effectiveness of the suggested approach.

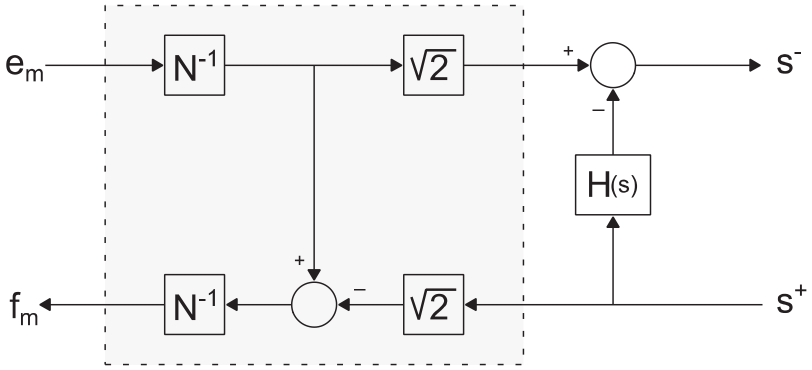


Figure 3.4: Geometric scattering scheme with wave filter

3.5 Conclusions

In this chapter, the requirements for a reusable control architecture for bilateral teleoperation were reviewed and a list of ideal criteria was established. Based on these targets, the choice for using specific design strategies (i.e. passivity based control) and interfacing was determined. For the controller architecture the wave variable method will be used for coding signals during transmission over the communication line, since stability can be guaranteed in this case. In order to provide stable interaction, the IPC method will be implemented on both sides of the channel, for providing stable interaction between user and haptic device, and telerobot and environment. Additional position information will be sent across the communication line in order to provide absolute position control for minimizing drift and tracking errors. Finally, in order to provide additional filtering for the purpose of eliminating undesired disturbances from non-idealities and modeling errors, wave filtering will be employed. The next chapter will discuss the design of these control methods in more detail.

4 Design

Based on the background information provided in chapter 2, in chapter 3 an analysis was put forth, where the key issues and main goals were considered. Several control solutions were addressed and selected in order to be able to satisfy the most essential design criteria. This led to a proposal for a design consisting of a particular interface strategy and controller design, in order to guarantee stable performance for a wide variety of bilateral teleoperation systems. This chapter addresses the development of the proposed control architecture yielding a modular design. The proposed architecture aims to be flexible enough to at least provide *basic* functionality to a wide spectrum of teleoperation systems; i.e. guaranteeing stable and acceptably operable behavior of the teleoperator, both in presence of communication delays and in case of interaction. Also, several performance enhancing control methods are considered for the purpose of improving accuracy and minimizing disturbances.

4.1 General design

Prior to designing specific control blocks on a lower level it is important to define a general structure for the complete system. Following the opted control methods presented in section 3.4, the wave variable control and Intrinsically Passive Control (IPC) frameworks are incorporated into a component-based modular architecture, capable of coping with delays and interaction respectively. The general structure of the teleoperation system, with the master and slave controllers expanded, is presented in Figure 4.1. Here, the control elements which are implemented according the same control concept are colored identically, and the local and remote sites are indicated by red and blue respectively. Also, the behavior and application specific control elements are marked.

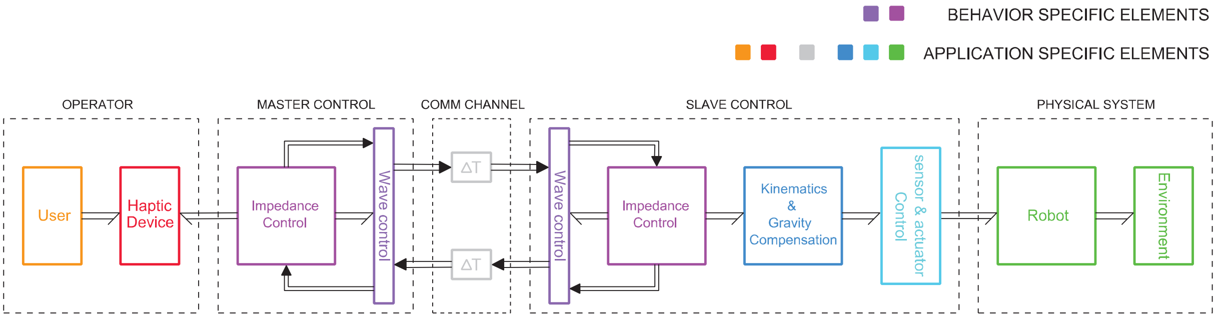


Figure 4.1: Block diagram of complete system

The behavior specific elements are the subject of main interest in this report with regard to reusability. With reference to Figure 3.1, this part of the system is constituted by the "*Impedance Control*" (in this case IPC; a type of impedance control) and "*Wave Control*" modules. Using a coordinate free representation, it should be possible to insert these modules without the need to consider a coordinate frame. In order to conform to the generalized interface that was covered in section 3.4.3, all IPC modules will be equipped with 6-DOF power ports, representing the energy exchange based on twists and wrenches. In addition, a one-directional position signal is transmitted along each module in the form of a 4×4 homogeneous transformation matrix. Also for the wave modules this interface is used, though the 6-DOF port on the 'wave side' of the module carries wave variables rather than power variables.

Since most controller designs require a least some knowledge of the system to be controlled, it is important to specify these demands. Precise knowledge of the communication medium and remote site are not required, as no predictive control elements are used in the design. In most cases this information is not available anyway since a typical communication medium such as the Internet is highly unpredictable, and in e.g. exploration applications little to no information about the environment or a remote objects' dynamics may be available. The proposed architecture does require knowledge of the telerobot kinematics and, for the purpose of gravity compensation, the masses of the links.

First, the next subsections will shortly cover the internal design of the blocks from Figure 4.1 that are *not* part of the control architecture designed throughout this report (i.e. the application specific elements), though for simulation purposes require a reliable representation. Subsequently, sections 4.3 and 4.4 will cover the design of the behavior specific modules in detail, respectively the wave control modules and impedance/IPC modules.

4.2 Application specific elements

This section will address the blocks and control modules specific to a certain application. This means that these blocks, as was argued in section 3.3.1, need to be redesigned to a certain extent with every other application.

4.2.1 Operator; user and haptic device

The operator block in Figure 4.1 consists of the "*user*" (the human operator) and the "*haptic device*". The haptic device in this case has been represented by the 'Geomagic Touch' (see Figure 4.2), which is also used with the youBot setup present at the University of Twente, and characterized by the following specifications (Table 4.1):

M_h	$0.045\ kg$
$B_{backdrive}$	$< 0.26\ N$
F_{max}	$3.3\ N$
$F_{max,continuous}$	$0.88\ N$
K_x	$1.26\ N/mm$
K_y	$2.31\ N/mm$
K_z	$1.02\ N/mm$

Table 4.1: Geomagic Touch specifications

Figure 4.2: The Geomagic Touch

As will be elaborated on in section 5.2.1, the dynamics of the haptic device have an important effect on the allowed controller settings for the IPC. Due to its low mass, high frequencies can arise in the system, which puts a limit on the attainable bandwidth of the controllers. However, during operation the haptic device is typically coupled to the human hand, and thus this effect will be less significant. Nevertheless, the user can let go of the device, changing the operator dynamics, possibly causing undesired effects. To this end, in this research the user will be modeled as a force only (albeit filtered), acting on the haptic device with given specifications.

4.2.2 Communication channel

The "*COMM channel*" in Figure 4.1 represents the delayed medium by which information is communicated between the master and slave sites. This could be a simple constant delay or a more complex element such as a typical internet connection with varying delay, missing information due to packet loss, and other imperfections. However, practically all modern control systems operate from a digital environment. In fact, the presented architecture is itself of a digital nature and therefore, without significant loss of generality, it is assumed that the information transmitted over the communication line will always be of a discrete nature. Though the physical line itself behaves as a continuous element, the effects of the line on the system will be limited to discrete varying delays, loss of packages and possibly disconnections. In chapter 5, the exact content of the delay blocks will be addressed, most importantly the types of delays that occur and their effect on the system.

4.2.3 Use-case: the youBot

The "*Physical System*" consists of the *robot* and remote *environment* (i.e. the interactions with base floor and arm end effector). As use-case for this report, the *youBot model* is used as the telerobot. The model is based on the design from (Dresscher et al., 2012) and includes both the omnidirectional platform (Figure 4.3.a) and the 5-DOF arm (Figure 4.3.b). Since the assignment is concerned with a simulation exercise, this model will serve as the virtual representation of the physical youBot with which a user interacts.

Figure 4.3: the KUKA youBot omnidirectional platform **b. the youBot 5-DOF arm**

(source for both Figures: (Locomotec, 2010))

Furthermore, the blocks labeled *"sensor & actuator control"* in Figure 4.1 represent the conversion element for data from the actual sensors and actuators of the robot to joint space (i.e. to angular velocities and torque setpoints), and vice versa. For the youBot setup the necessary computational models are readily available, and as such are not a subject of study in this assignment, moreover since for simulations these are not needed.

From a control point of view, there are several ways to control motion and interaction of the entire youBot (i.e. the combination of base and arm). A first approach is by considering both the base and arm as being a single kinematic structure with one end-effector, operated from a single haptic device. This would either imply determining actuator commands for the base and arm joints based on a single task frame (e.g. the gripper), or alternatively by switching between task frames hence controlling base and arm separately and one-at-a-time. Another approach is to consider both the base and arm independently, possibly using two haptic interfaces. As suggested by the general structure presented in Figure 4.1, the youBot arm and base are controlled independently in this research, and for the remainder of this report this will be the default setup. This does not only simplify the design (without loss of generality), it also facilitates the opportunity to verify the reusable nature of the generic (i.e. behavior specific) modules; separately on the base and arm.

4.2.4 Kinematics & gravity compensation

In order to perform a transformation from joint space to a geometric representation, kinematic models are inserted between the robot joint interfaces and the rest of the system (in Figure 4.1 the *Kinematics & Gravity Compensation* block). Due to the exact match between the model and 'real' youBot kinematics, both models will show identical motion for equal input commands. Clearly, in reality a perfect model of a physical system cannot be achieved, but in this case it does give a clear picture of the effects of certain system dynamics without being overshadowed by non-idealities.

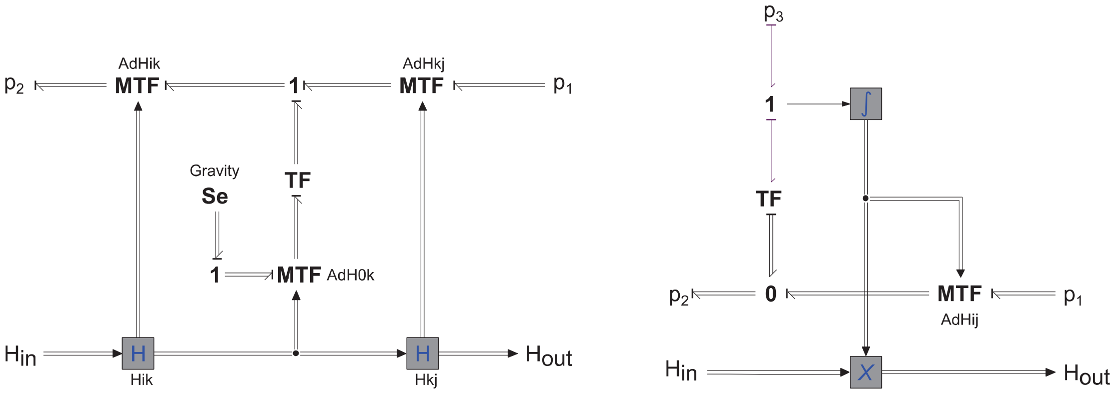


Figure 4.4: **a. Submodel of Link kinematics** **b. Submodel of Joint kinematics**

In any controller design, robustness to disturbances is an important criterion. A particular disturbance which can be easily dealt with without using complex algorithms is gravity. The most effective and straightforward way to implement compensation is by placing a controller which directly generates torque commands to the joints, in close proximity to the robot (hence not being affected by communication delays or other interfering system dynamics). Having required knowledge of the approximate mass of each link, the kinematics model can easily be extended with gravitational effects applied directly to the CoG (center of gravity) of each link. The combined kinematics and gravity compensation block for the arm is composed of link and joint modules similar to those in the complete youBot arm model but without additional body dynamics, see Figure 4.4 a. and b.

The resulting model can be modified to output force commands to the joints by inverting causality to *effort-out*, such that the combined torques due to gravity and controller commands are outputted at the joint connections. For the arm this is straightforward with the provided model, but inverting causality for the youBot base model is a more challenging task.

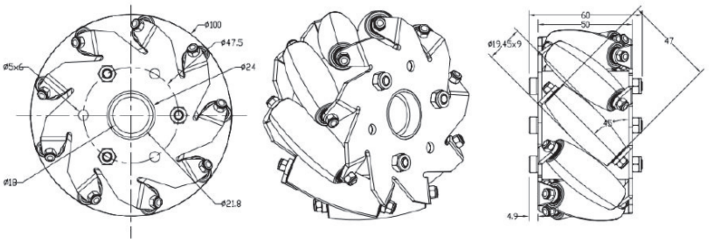


Figure 4.5: A design for a typical Mecanum wheel ***(source: robocave.pk)***

The base model from (Dresscher et al., 2012) does not allow inverting causality of the joints, inherently due to the structure of the design. Specifically the model for the mecanum wheels does not allow reversing signal direction for flow and effort due to the fact that it discards velocity information in the transformations for the rollers of the wheel (see Figure 4.5). To that end the model for the mecanum wheels has been altered such that such that it can be used as kinematics element, of which the design can be found in Appendix D. Clearly though, it is not possible to account for the dynamics between youBot and floor such as friction and slipping, using purely a kinematics model. This means that certainly there will arise steady-state errors during operation between the kinematics model (which is coupled to the slave position) and the physical youBot.

4.3 Wave variable control

As was outlined in section 2.3 and motivated in section 0, the wave variable method will be implemented as a means to provide stable communication over a delayed transmission line connecting the local and remote subsystems. Due to the fact that wave variables are by definition insensitive to delays makes it a very robust control strategy for teleoperation over a delayed communication line. However, as noted before, there are limitations to the wave variable design. Several studies indicate performance degradation with increasing delay, and task completion time increasing significantly (Ching and Book, 2005). This section will initially provide a straightforward implementation of the wave variable design and will then focus on additional elements for further enhancing system performance.

4.3.1 Design and interfacing

Section 2.3 has discussed the evolution from scattering variables toward a more general, multidimensional framework similar to what is being used in the current work. Note though, that while not necessarily taking part in the actual scattering transformation, the additional homogeneous matrix transmitted along the line is included in the interface. Though the coordinate free representation allows for connection to any arbitrary geometric framework (provided it operates in the same dimension space) all elements in the controller design are bounded to real world (three-dimensional) space and will eventually adopt the coordinate frame of the system they are being connected to. In fact, due to the additional position information transmitted parallel to the line, the coordinate frame becomes instantly defined with respect to a certain 'world frame' when being connected to another subsystem. Hence, any initial conditions must be defined using these position signals, either by setting initial values or by synchronization.

As was pointed out in section 2.3.2, neither of the multi-DOF approaches from equations (2.6) and (2.10) addresses the complete family of passive scaling matrices. Additionally, it was recognized in (Alise et al., 2009) that both formulations can be related to one another by setting:

$$A_w = C_w = \boldsymbol{N}^{-1}/\sqrt{2} \qquad \text{and} \qquad B_w = D_w = \boldsymbol{N}/\sqrt{2} \tag{4.1}$$

Conversely, in order to evaluate generality of the geometric scattering transformation this can be rewritten as:

$$\boldsymbol{N}^{-1} = \sqrt{2}A_w = \sqrt{2}C_w \qquad \text{and} \qquad \boldsymbol{N} = \sqrt{2}B_w = \sqrt{2}D_w \tag{4.2}$$

Using $(\lambda\boldsymbol{N})^{-1} = \lambda^{-1}\boldsymbol{N}^{-1}$ in which λ is a scalar and N a square matrix (which is valid since $\boldsymbol{N}$ is invertible and $\lambda \neq 0$) it holds that $\sqrt{2}A_w = \left(\sqrt{2}B_w\right)^{-1} = B_w^{-1}/\sqrt{2}$, which implies that for the geometrical scattering case that:

$$A_w = C_w = \frac{1}{2}B_w^{-1} = \frac{1}{2}D_w^{-1}$$

With regard to the conditions provided in (2.11), these equations show that the geometrical scattering representation allows only a very restricted class of scaling matrices, more specifically the special case in which $Q = I,\ S_w = 0$ and $A_w = A_w^T$ (meaning A_w is restricted to be a symmetric matrix). For an intuitive idea how these restrictions influence the signal flows, it can be clarifying to consider the consequences schematically. A schematic representation for the multi-DOF equations in (2.6) is shown in Figure 4.6.

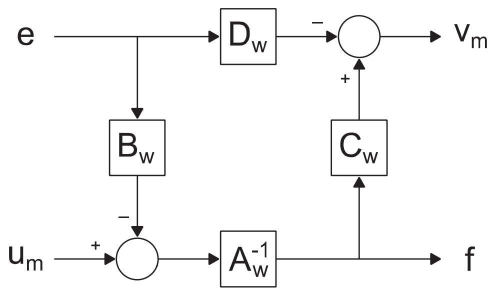


Figure 4.6: Schematic representation for multi-DOF wave variable mapping

Using the relations from (4.1) and (4.2) this block diagram can be rearranged (according the in Appendix C provided schematic) to yield the geometric scattering transformation as is reported in Figure 4.7.

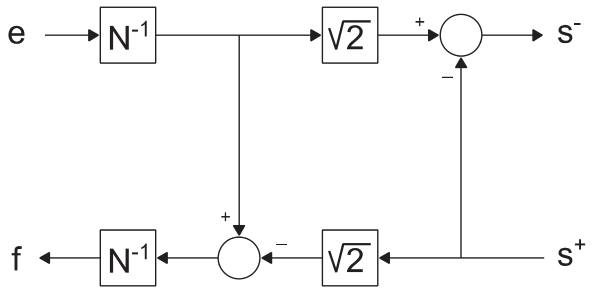


Figure 4.7: Signal diagram for geometric scattering transformation

Whether the restrictions of the geometric design put a significant limit on the achievable generality with respect to performance remains an open question. Based on experiments from (Alise et al., 2009) with a two-DOF linear model, the effect of the three matrices used in equations (2.11) on system performance can be characterized as follows (as already mentioned in section 2.3.2 from):

1) No matter which Q is used, the overall system output will not change.
2) As A_w changes, there is a tradeoff between speed and accuracy. In experiments users performed worse when A_w had larger terms off the main diagonal, reflecting to larger amounts of cross-coupling.
3) As the norm of S_w increases, the master and slave tend to experience more overshoot and longer settling time.

This suggests that the consequences for the geometrical design may actually be not too restrictive; if the choice of Q is irrelevant to the system output then $Q = I$ is as good as any other value — if a lower value on the norm $\|S_w\|$ is beneficial for the response then with $S_w = 0$ the minimum norm is achieved — then only the symmetry restriction on A_w will affect generality. Moreover, due to the claim in point 2) the latter restriction is only a minor limitation, since terms off the main diagonal of A_w have a negative influence on performance. However, Q does change the wave variables that are sent across the communication line by rotating the separate components relative to each other. As such, the waves themselves may actually be affected by control elements in the wave domain, possibly influencing the output after all. Also, an experiment with a 2-DOF (x, y) system showed that a nonzero value for S_w improved position tracking and reduced steady-state error for one variable (x) at the cost of only slightly increasing the steady-state error of the other variable (y), see (Alise et al., 2009). Since the geometric scattering design relates to a six-dimensional signal transmission, there is no easy way of predicting the effect in such a design, and the possibility to use other values for S_w may be beneficial for some systems. Finally, in cases where values off the main diagonal for A_w are desired for whatever reason, it may be desirable to not having to mirror the value to the other half of the main diagonal.

These considerations raise the question whether, in light of generality, the three-matrix (3M for convenience) design is not beneficial after all, since providing more tuning parameters yields a bigger operational range. Unfortunately, this would involve major changes to the proposed architecture which may negatively affect reusability of the modules. Clearly, it is by all means possible to incorporate the design and still use a bond graph interface, since after all; decoding the wave variables will in both cases yield conjugated power variables. However, impedance matching and scaling becomes a far more time-consuming task. When comparing Figure 4.6 with Figure 4.7 it is clear that there is no way to implement the 3M-design in a bond graph design for other values than those allowed for a $\mathbf{TF}$ element (i.e. a transformation ratio $\boldsymbol{N}^{-1}$ in case of geometric scattering).

Though it might actually be possible to incorporate the A_w, S_w and Q matrices design into the architecture without compromising too much the generality, (formally) characterizing the effect of these scaling matrices on performance for teleoperation systems in general would be no less than a study on itself. Moreover, instead of having to tune one parameter ($\boldsymbol{N}^{-1}$) an operator would have to weigh off three independent parameters, hence increasing operating complexity. It is at least questionable whether this outweighs the benefits of additional tuning options, certainly since their effect, as was reasoned earlier, is limited. Finally, the geometric coordinate free representation allows for connection to any arbitrary coordinate system, which is precisely the beauty of this kind of architecture and which makes it such a valuable addition to an intended reusable design.

With the previous considerations in mind, the geometric design will be adopted and the limitations of this design decision can be evaluated rather than modifying the essential part of the geometric wave module. Using these arguments it is now possible to define the basic, straightforward design for the wave module following the geometric scattering approach given in (Stramigioli et al., 2000) governed by the equations in (2.10). These equations can be rearranged so as to express the outgoing wave $\boldsymbol{s}^-$ as function of the incoming wave $\boldsymbol{s}^+$ and a power variable. Conversely, either of the power variables $\mathbf{e}$ or $\boldsymbol{f}$ can be expressed as a function of their power conjugated dual and the incoming wave $\boldsymbol{s}^+$. This yields a total of two sets of possible wave equation pairs, governed by the following equations (for completeness the equation for $\boldsymbol{s}^+$ has also been reported):

$$\begin{aligned} \boldsymbol{s}^+ &= \sqrt{2}\,\boldsymbol{N}\,\boldsymbol{f} \,+\, \boldsymbol{s}^- & \qquad \boldsymbol{e} &= \boldsymbol{Z}\,\boldsymbol{f} \,+\, \sqrt{2}\,\boldsymbol{N}\,\boldsymbol{s}^- \\ \boldsymbol{s}^- &= \sqrt{2}\,\boldsymbol{N}^{-1}\,\mathbf{e} \,-\, \boldsymbol{s}^+ & \qquad \boldsymbol{f} &= -\boldsymbol{Z}^{-1}\,\boldsymbol{e} \,+\, \sqrt{2}\,\boldsymbol{N}^{-1}\,\boldsymbol{s}^+ \end{aligned} \tag{4.3}$$

As was pointed out in section 2.3.3, it has been shown that there are multiple reasons for choosing the flow as an output variable on the power side at both sites. Consequently, computing $\boldsymbol{f}$ and $\boldsymbol{s}^-$ as a function of $\boldsymbol{e}$ and the incoming wave variable $\boldsymbol{s}^+$ will be the standard way for computing the wave variables at both the master and slave sides, hence modularizing the design is rather straightforward. Moreover since, in order to create a perfectly symmetrical system, the causality at both ports of the communication line should be the same. The lower two equations in (4.3) can be expressed schematically as presented in Figure 4.8, where the I-block performs the transformation from power variables to waves and vice versa and the $\mathbf{MTF}$ element represents the modulated transformer with transformation ratio $\boldsymbol{N}^{-1}$. This ratio holds a strong connection to the impedance of the rest of the system as has been addressed in section 2.3.2.

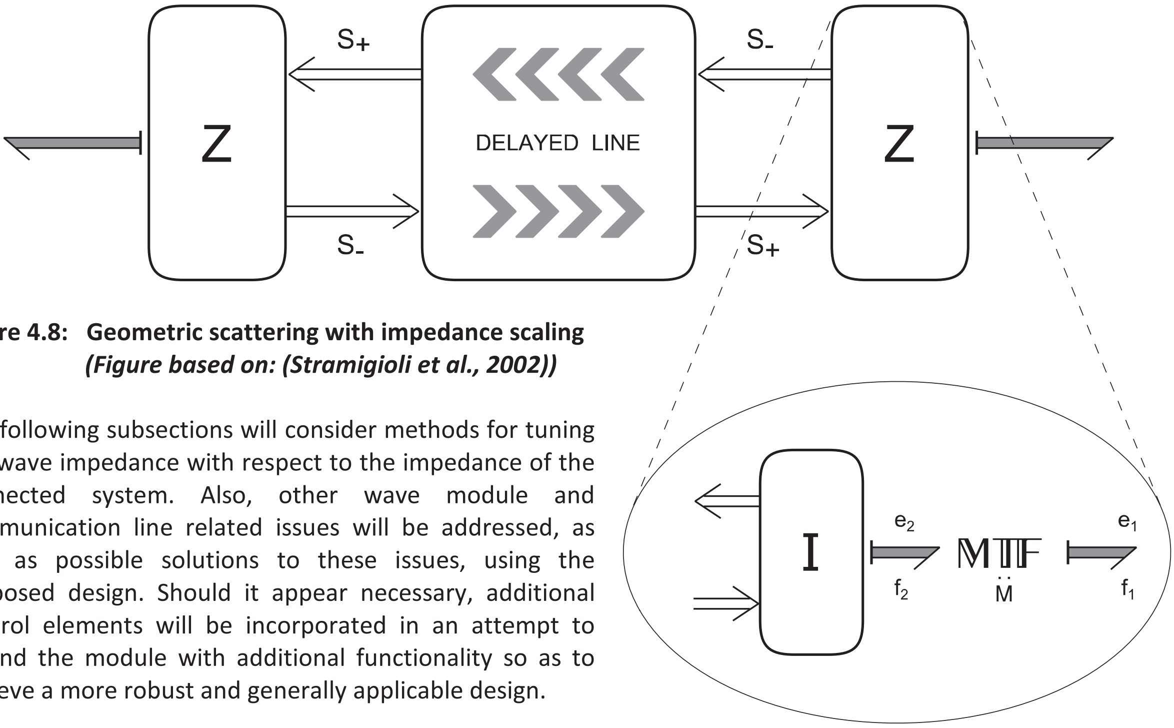


Figure 4.8: Geometric scattering with impedance scaling
(Figure based on: (Stramigioli et al., 2002))

The following subsections will consider methods for tuning the wave impedance with respect to the impedance of the connected system. Also, other wave module and communication line related issues will be addressed, as well as possible solutions to these issues, using the proposed design. Should it appear necessary, additional control elements will be incorporated in an attempt to extend the module with additional functionality so as to achieve a more robust and generally applicable design.

4.3.2 Wave filtering

In section 3.4.5 it was argued that in order to provide filtering in the communications, while at the same time preserving as much of the original signal information as possible in terms of frequency spectrum, a wave filter can be strategically placed in the communication loop. The resulting scheme (when applying the wave filter only in the wave feedback path on the master side) is displayed in Figure 3.4.

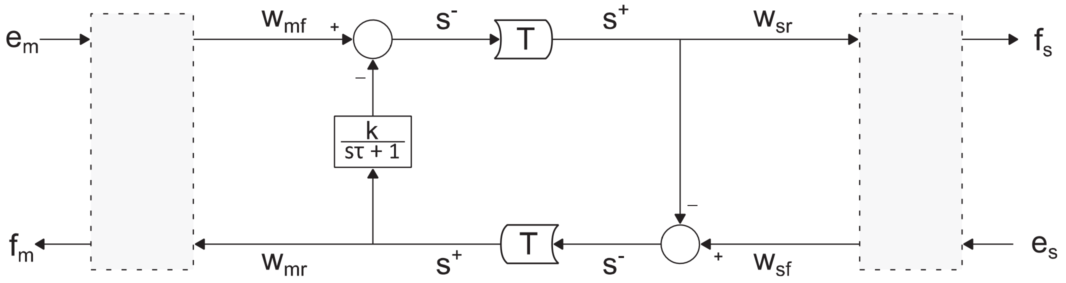


Figure 4.9: Communication line with delay and wave filter only on master side wave feedback loop

The loop equation for this scheme can be computed straightforwardly as follows (writing $H(s)$ for $k/s\tau + 1$):

$$W_{sr}(t) = W_{mf}(t - T) - H(t)\left(W_{sf}(t - 2T) - W_{sr}(t - 2T)\right)$$

with W_{mf} the scaled effort ($W_{mf} = \sqrt{2}\boldsymbol{N}^{-1}e_m$) at the master side, W_{mr} the incoming wave at the master side, W_{sr} the incoming wave at the slave side and W_{sf} the scaled effort at the slave side, see Figure 4.9. When the delay of the channel tends to zero, the behavior of the communication loop in the forward path becomes:

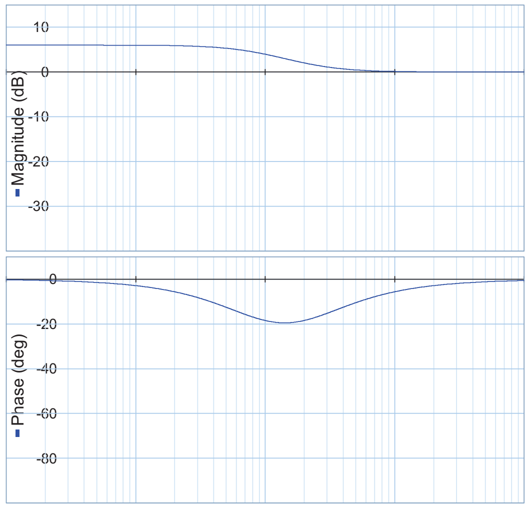
$$W_{sr} = W_{mf} - H\left(W_{sf} - W_{sr}\right) = W_{mf} - H\,W_{sf} + H\,W_{sr}$$

$$(1 - H)\,W_{sr} = W_{mf} - H\,W_{sf}$$

$$W_{sr} = \frac{W_{mf} - H\,W_{sf}}{1 - H} = \frac{W_{mf}}{1 - H} - \frac{H\,W_{sf}}{1 - H}$$

Performing a Laplace transformation on the equations and rewriting, using a first order filter of the form $H(s) = k\ /\ s\tau + 1$, yields a transfer function for the incoming wave on the slave side:

$$W_{sr}(s) = \frac{s\tau + 1}{s\tau + 1 - k}W_{mf}(s) - \frac{k}{s\tau + 1 - k}W_{sf}(s) \tag{4.4}$$



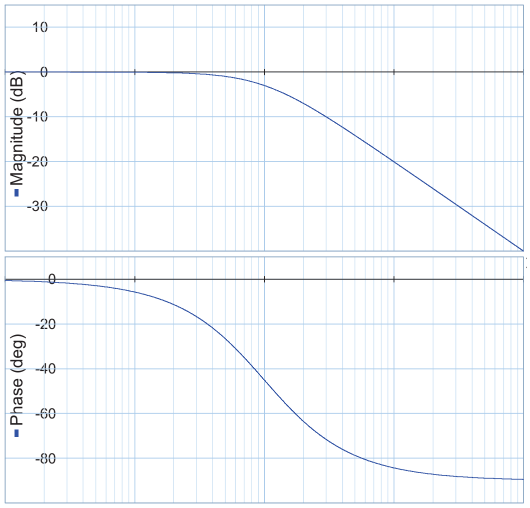


Figure 4.10.a: Transfer of $H(s) = \frac{(s\tau+1)}{(s\tau+1-k)}$ **b. Transfer of $H(s) = \frac{k}{(s\tau+1-k)}$**

As was noted in section 3.4.5, the gain k must be chosen ≤ 1 in order to avoid instability, which also clearly follows from the above equations, since for values with $k > 1$ the pole of the filter shifts to the right half of the pole-zero plane. The transfer behavior for both terms in Equation (4.4) can be found in the bode plots of Figure 4.10.a (first term) and Figure 4.10.b (second term). The parameters used are $\tau = 0.05$ and $k = 0.5$, hence resulting in a crossover frequency at $f_c = 10\ Hz$, which is in accordance with the comment from section 3.4.5 with regard to maximum frequency content in human hand generated signals.

In a similar fashion the behavior for of the channel from slave to master (i.e. the return path) can be deduced, which yields the following transfer function for the incoming wave at the master side:

$$W_{mr}(s) = \frac{s\tau + 1}{s\tau + 1 - k} W_{sf}(s) - \frac{s\tau + 1}{s\tau + 1 - k} W_{mf}(s) \tag{4.5}$$

From the results reported in equations (4.4) and (4.5) and Figure 4.10 it is clear that only the path from W_{sf} to W_{sr} becomes low-pass filtered. Unfortunately, with the current settings the remaining three paths are also altered, showing low frequency gain and some phase lag around 15 Hz, which may possibly result in undesired behavior. Whether this has a significant influence on the system will have to become clear during experiments.

Substituting the relations for e_m, f_m, W_{mf} and W_{mr} into the equations, i.e.:

$$W_{mf} = \sqrt{2}\boldsymbol{N}^{-1}e_m \qquad W_{sf} = \sqrt{2}\boldsymbol{N}^{-1}e_s$$

$$f_m = \boldsymbol{Z}^{-1}e_m - \sqrt{2}\boldsymbol{N}^{-1}\, W_{mr} \qquad f_s = \boldsymbol{Z}^{-1}e_s - \sqrt{2}\boldsymbol{N}^{-1}\, W_{sr}$$

and similarly for the slave side, yields the following relations for the complete system from Figure 4.9:

$$\begin{aligned} f_s &= \boldsymbol{Z}^{-1}e_s - \sqrt{2}\boldsymbol{N}^{-1}\left(\frac{s\tau + 1}{s\tau + 1 - k}\sqrt{2}\boldsymbol{N}^{-1}e_m - \frac{k}{s\tau + 1 - k}\sqrt{2}\boldsymbol{N}^{-1}e_s\right) \\ f_m &= \boldsymbol{Z}^{-1}e_m - \sqrt{2}\boldsymbol{N}^{-1}\left(\frac{s\tau + 1}{s\tau + 1 - k}\sqrt{2}\boldsymbol{N}^{-1}e_s - \frac{s\tau + 1}{s\tau + 1 - k}\sqrt{2}\boldsymbol{N}^{-1}e_m\right) \end{aligned} \tag{4.6}$$

From this it is clear that the position signal at the master, in this configuration, contains the undamped force components from both the master and slave. On the slave side, the position signal is reconstructed from the undamped force from the master as well, but the wave filter does create high frequency damping to the slave side force. This conclusion may seem contra-intuitive since the wave filter is placed at the master side. However, due to the filter being placed in the feedback path at the master side, as viewed from the slave, the effort signal from the slave which travels the communication line is the only direct feedback path being filtered, see Figure 4.11. Any signals traversing the inner loop are filtered and hence the wave filter creates continuous damping for higher frequent signals. Simulations in chapter 5 will analyze the effectiveness of this design.

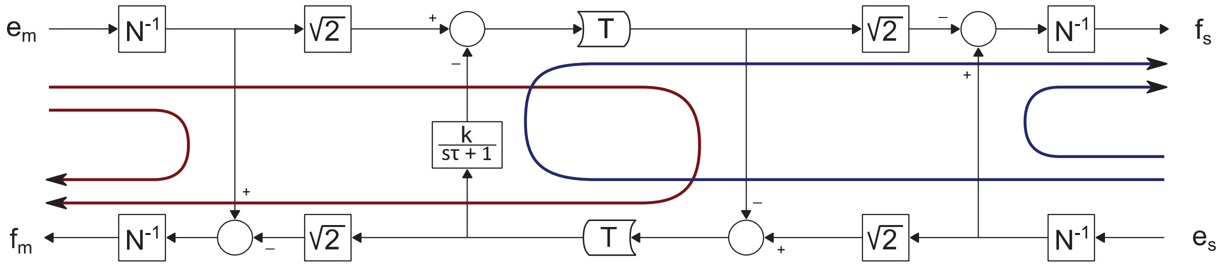


Figure 4.11: Feedback paths from master and slave crossing the communication line.

4.4 Intrinsically Passive Control

In section 4.3, the wave variable method has been applied primarily as a means to guarantee *stability* of the overall system. The main reason for using the IPC framework is to guarantee stable *interaction* between the remote operator, haptic device, telerobot and remote environment. The homogeneous position information at the interfaces can be used, as was argued earlier, for synchronizing master and slave systems at start-up. Also, the IPC modules can be employed to provide direct position control, as a means to obtain an acceptable level of *accuracy,* as will be addressed in section 4.4.2. In order to be able to quickly implement the several IPC-elements yet provide enough flexibility to a designer, the IPC design has been split into modules that describe physically interpretable dynamic behavior. The next sections will cover the internal design of these modules.

4.4.1 IPC Modules

The most straightforward way to split the design is by using modules which describe *spatial compliances* and modules which represent *virtual objects*. With regard to the operator or designer this is an intuitive way to section the controller and tune the dynamical appearance of the system, since these elements are physically interpretable and directly influence way the operator perceives the system. Tuning the *compliance properties* characterizes the stiffness of the system and thus affects the force needed to maintain a relative displacement of the master virtual object. Since telemanipulation deals with delays this is a common occurrence; when the operator moves the local robot (e.g. haptic device) the remote robot will not follow instantly, creating a distance between the master and slave objects. The compliance then, influences the feedback force felt by the operator. Changing the *virtual object properties* characterizes the inertia and hence determines how much effort it takes to achieve a certain velocity of the virtual mass, and consequently the telerobot. Finally, changing the *damping properties* will ensure settling of the system and reduce oscillations. Figure 4.12 shows the design using these elements, where on either side of the delayed communication channel (T) a combination of compliances and masses represents the IPC (i.e. the "*Impedance Control*" elements from Figure 4.1).

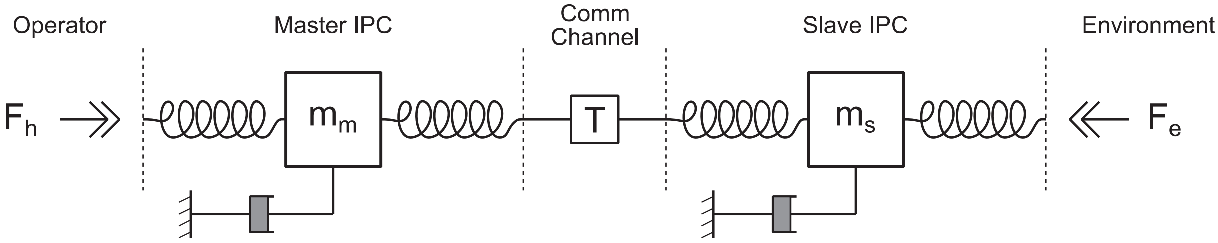


Figure 4.12: IPC controllers connected through time delay

The reason for this particular setup is due to the fact that the wave modules output a velocity at the power side and hence behave as an admittance port (a direct feedback loop with impedance $\boldsymbol{Z}^{-1}$; recall Figure 4.7). To that end, any control element being connected to the power port of a wave module should have effort-in causality. With respect to the IPC modules this means that no virtual object can be directly connected to the wave module due to its fixed flow-out causality. Hence, the spatial compliance is the only IPC module available for connection to a wave module. In order to provide damping to the system, damping elements should be added to the design. However, damping can only be injected at virtual objects. Applying a damping element between the local or remote site and the communication channel would result in a causal inconsistency, since the velocity information in that case directly depends on the operator or telerobot force. Solving this issue would require a measurement of the velocity, which may not be available in a lot of applications. To this end, both on the master and slave sites the controller incorporates a damping element inside the virtual objects only.

4.4.2 Tracking and drift

Transmitting position information along the signal path provides the possibility for improving tracking accuracy and minimizing drift, using any of the methods discussed in section 3.4.3 (and reviewed in Appendix F). By calculating the position difference between master and slave virtual objects, the residual spatial distance (i.e. offset between master and slave) can be eliminated. This can be done by passively changing rest length, or possibly the stiffness, of the remote and/or local compliances (recall section 2.4.4 on varying stiffness rest length) so as to compensate for any resulting offset. During operation, the master and slave objects are then 'pulled' toward each other. In the one-dimensional case this can be represented using $F = -k \cdot x + x_{diff}$ where, in the geometric case, x_{diff} is the homogeneous transformation matrix H_m^s, with m the master frame and s the slave frame.

4.4.3 Spatial compliance module

The compliance module should generate a restoring force between two rigid bodies based on their relative distance, which is calculated from a relative twist. From the controller viewpoint, both the local site (operator and haptic device) and remote site (telerobot) are viewed as an inertia. Consequently, the compliance module will have a *flow-in* causality at both ports from which a relative twist is calculated, which by definition means that both ports will output an *effort* (i.e. a wrench). Since both ports accept a twist from which the difference is directly calculated, the output wrench at these ports must be opposite but equal in absolute magnitude, which is an obvious result of to the nodic behavior of a compliance element. Because springs are deformed when being subjected by a wrench, the endpoints will have a relative position *and* orientation with respect to each other (i.e. both a translation and a rotation component). Alternatively, with the proposed architecture the homogeneous transformation matrix can be used to express the spatial distance between the two endpoints.

The schematic design for the spatial compliance is presented in Figure 4.1, where $\mathbf{C}$ element represents the spatial compliance, as defined by Equation (2.12), the $\mathbf{0}$ port performs the summation of twists, and the $\mathbf{MTF}$ elements provide coordinate transformations of the twists and wrenches. The signals originating at the $\mathbf{1}$ ports carry the respective twist to the $\mathbf{MTF}$ element and integrate this twist into a homogeneous position matrix. Finally, the multiplication block calculates the homogeneous position difference from one side of the spring to the opposite side. The $\mathbf{MSf}$ element receives position information (input Trl) from the virtual object for the purpose of direct position control.

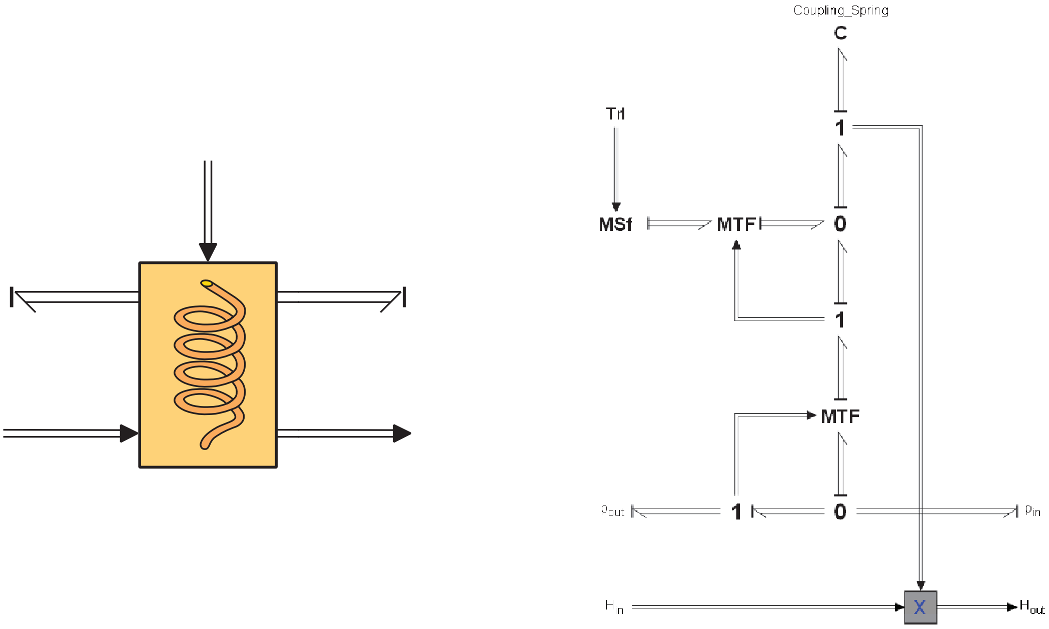


Figure 4.13 : The spatial compliance module

The presence of the $\mathbf{MTF}$ elements is required due to the relative distance between the extremities of the spring when it is being deformed. Consequently, a change of coordinates is necessary to correctly express the twists and wrenches on both sides of the element. In order to provide this mapping to the appropriate coordinate frames, a modulated transformer is incorporated with a transformation ratio equal to the adjoint of the relative distance, i.e.:

$$M = Ad_{H_i^j} := \begin{pmatrix} R_i^j & 0 \\ \tilde{p}_i^j R_i^j & R_i^j \end{pmatrix}$$

The stiffness of the compliances will determine how the user experiences the interaction with the system. Since for perfect transparency the stiffness between the operator and remote robot should be $K = \infty$, the stiffnesses in the line are preferably tuned as high as possible. A too low stiffness would result in a 'mushy' feel and slow response. However, a too high stiffness will cause high frequent oscillatory behavior if there is not enough damping present in the channel. To that end, a certain amount of damping should be injected into the system (i.e. in the virtual object module, for the reasons addressed in section 4.4.1).

4.4.4 Virtual object module

Dual to the compliance design, the virtual object block calculates the twist of a geometrical mass due to forces acting on it. As a result, the virtual block module accepts multiple wrenches from which calculates a resultant wrench which is integrated into a body momentum, and a twist is outputted. As such the module has an *effort-in* and hence *flow-out* causality at its ports. Though it would be possible to connect more than two subsystems to the virtual object (e.g. in a grasping application, see (Stramigioli et al., 1999)) initially the virtual object is implemented as a two port since it will be placed in the bi-directional teleoperation path. The resulting schematic design of the virtual object module is given in Figure 4.14 where the $\mathbf{I}$ element implements the virtual object as defined by Equation (2.14), the $\mathbf{1}$ port performs the summation of wrenches and $\mathbf{R}$ is the damping element in order to guarantee steady-state convergence, specified in Equation (2.15). This element provides damping relative to the world frame, as indicated by the $\mathbf{Sf}$ element which outputs a zero flow. For the remainder of this report, all representations of the virtual object module are implemented with such a damping element. As may be recalled from section 2.3.3, the damping value should be chosen with care, since in combination with the wave impedance $\boldsymbol{Z}$, this parameter is decisive for providing impedance matching and eliminating reflections. Finally, the position information from the virtual object is transmitted from the master or slave virtual objects and can be used as input to the $\mathbf{MSf}$ port of the spatial compliance (Figure 4.13) on the other side of the channel.

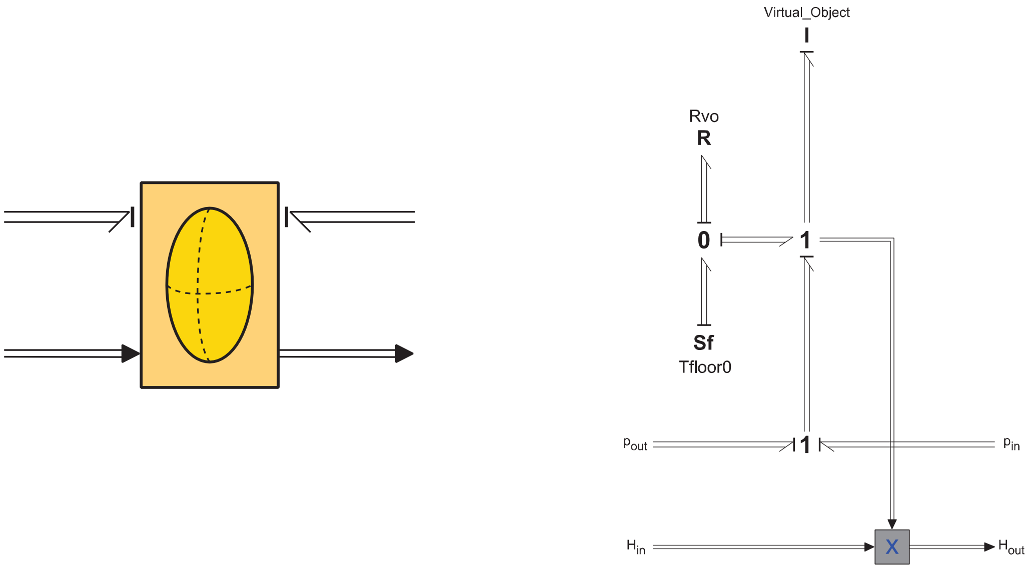


Figure 4.14 : The virtual object module

4.5 Conclusion

In this chapter the control methods discussed in chapter 3, for use in the teleoperation control architecture, were evaluated and realized. For the use of transmitting information across a delayed communication line, the wave variable modules were developed. Also, in order to provide some filtering in the wave domain, which can reduce disturbances in a passive way, while maintaining valuable frequencies, a new way of placing such a filter was developed. Finally, the modules for the IPC controller were designed, providing the necessary means for direct position control in order to improve accuracy. The next chapter will evaluate the controller design by performing simulations.

5 Simulations

In this chapter, performance of the proposed control architecture is analyzed. First, simulations are performed to verify the introduced control strategies separately, using simple models. Also, the effect of tuning the control parameters on system behavior is evaluated. In the final section, the results of simulations performed with the youBot are presented, using the complete control architecture proposed throughout chapters 3 and 4.

5.1 Model and experiments

This section addresses the requirements for a feasible general teleoperation system and how to experimentally verify the effect of the control elements on system behavior. The most important question to be answered is whether the design provides a general enough framework to serve as a platform for a typical teleoperation system, and whether behavior is acceptable with respect to the considerations provided in sections 3.2 and 3.2.1. The criteria presented in section 3.2.1 for an ideal teleoperation system are recapped in Table 5.1 and are used as a measure for quantitatively characterizing overall system performance.

CRITERIUM	GOAL
Stability	Always Stable
Inertia & Damping to Master	Inertia $M = 0$ & damping $B = 0$
Stiffness to Master	Stiffness $K = \infty$
Slave tracking	No error: $\delta = 0$
Drift between Master-Slave	No drift: $\Delta = 0$

Table 5.1: Criteria for ideal telemanipulation

Simulations are performed in order to comment on the overall system stability and whether indeed stable behavior can be achieved during different typical situations, and if not, what the constraints or conditions are. Where needed, several options for improving stability will be covered. Specifically with regard to the IPC modules, the effect of tuning the parameters for the spatial compliances and virtual objects will be evaluated. Finally, the accuracy of the complete system, characterized in terms of the tracking capabilities of the slave and drift between master and slave controllers, will be investigated. Additional control has been applied to improve accuracy, also in case of varying delays. The complete teleoperation system with controller design is depicted in Figure 5.1. All *generic* (application) elements from Figure 4.1 have been specified according section 4.2, in the form of a haptic device – *the Geomagic touch* – a specific telerobot – *the youBot* – and a communication line which will be addressed in subsequent sections. The *specific* (behavior) elements are the subject of study in this research; the "IPC" and "WAVES" control modules. Section 5.2 will focus on the IPC controller and section 5.3 on the wave variable controller ($\mathbf{Z}$-block) and investigate tuning of the parameter settings. Subsequently, the communication line will be introduced, thus creating the actual teleoperation setup, and the tracking and drift control elements will be introduced. Finally, the complete design will be evaluated using the complete youBot model, by first addressing the base platform and next the arm.

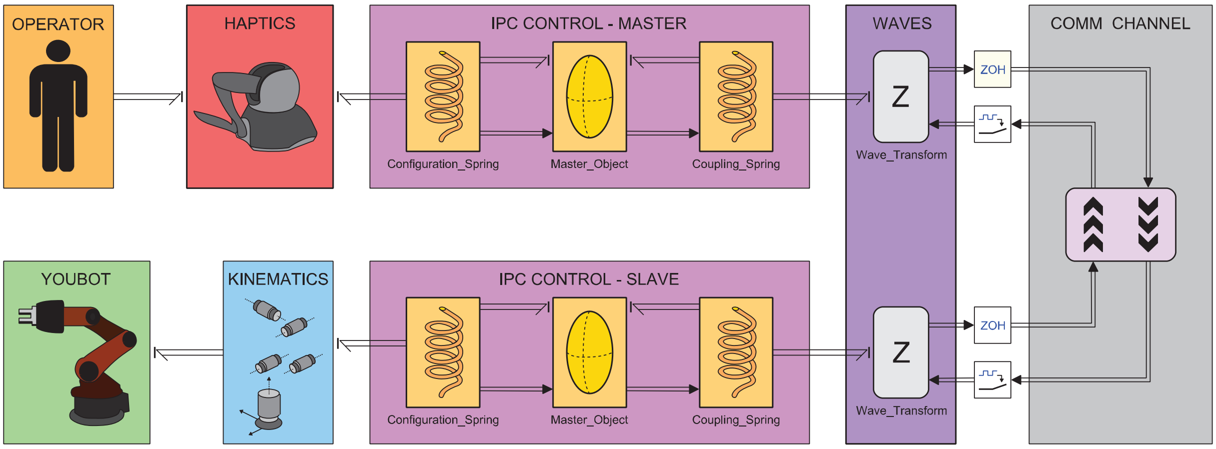


Figure 5.1: The complete teleoperation system controller design

5.2 Intrinsically passive control

This section addresses the use of the IPC modules and tuning of the control parameters. Though the elements are intrinsically passive from a theoretical point of view, there are certain factors (mainly due to the digital nature of the system) which pose a strict limit on the use of the control algorithm. The dynamic properties of the IPC design (stiffness, inertia and damping) directly shape the system behavior by influencing the natural frequencies. In order to investigate the behavior and potential of the IPC compliance and virtual object modules, in this section first a straightforward controller is implemented.

5.2.1 Master and slave controllers

In order to analyze the behavior of the master and slave controllers, first the actual implementation of the IPC modules is considered. As outlined in section 4.4, the design for the master and slave controllers consists of a spring-mass-spring configuration. In the situation where the master is disconnected from the transmission line, or similarly in case of an (infinitely) large communication delay, the active part of the teleoperation system will reduce to the system depicted in Figure 5.2. The system is directly connected to the world frame, since the communication line outputs a constant zero velocity in this situation. Note that this is not the case when the system is connected to an open line using a wave module; in that case the master controller 'sees' a damper, hence limiting the maximum attainable velocity. This system consists of the haptic device and master controller (here indicated by the 'IPC CONTROL' box). Damping is only included in the master object (recall section 4.4.1), and the haptic device is modeled using the haptic device specifications from Table 4.1.

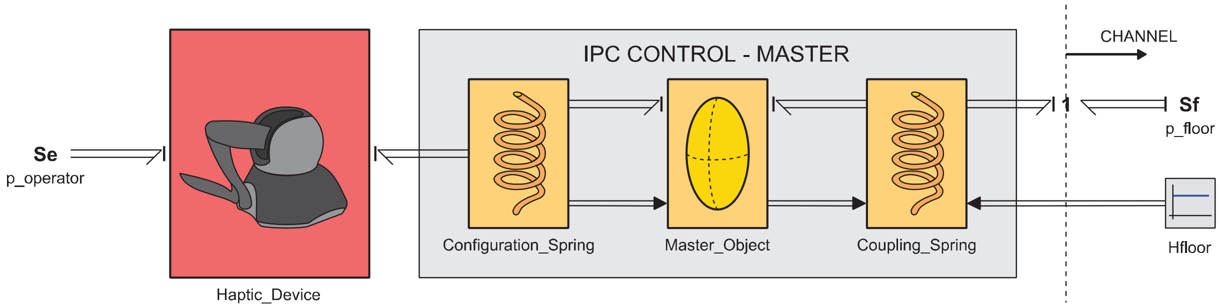


Figure 5.2: System connected to an open communication line (i.e. an infinite time delay)

In the ideal situation, applying a force to the haptic device will deform the spring, which then 'pulls' the master object toward the haptic device. In steady state, with no force applied, the coupling and configuration springs will return to their lowest energetic state (zero rest length). Using damping injection into either or both of the masses, both objects will converge to a stable position and coincide with each other. However, due to damping being present in the system, state-state offset may occur in the system. In section 5.4.1 a method for resolving this issue will be treated. Additionally, due to the discrete nature of the controllers, information is only available at sampling intervals, which limits the system bandwidth. For certain values of the IPC control elements, the overall system can in fact become unstable due to too high natural frequencies ($f_n > f_s$) being present in the system. To this end, careful tuning of the IPC modules is a necessity.

As was noted earlier, an as high as possible stiffness and an as low as possible mass are desirable for providing good transparency in a teleoperation system. In order to analyze the behavior of this system, choose K_{cp} as the coupling spring stiffness, M_m and B_m as the master virtual object inertia and damping respectively, K_{cf} the configuration spring stiffness, and M_h, $K_{x,y,z}$ and $B_{baxkdrive}$ as the haptic device properties. With regard to the criteria and desired goals (as defined in Table 5.1) and the haptic device specifications from Table 4.1, it follows that it must certainly hold that the $K_{cp} \gg K_{x,y,z}$ and $K_{cf} \gg K_{x,y,z}$ and also $B_m > B_{backdrive}$. Hence, the stiffness and damping properties of the haptic device are assumed negligible and the behavior of the system can be characterized by the following fourth order transfer function:

$$H(s) = \frac{X(s)}{F(s)} = \frac{\frac{M_m}{K_{cf}}s^2 + \frac{B_m}{K_{cf}}s + 1 + \frac{K_{cp}}{K_{cf}}}{\frac{M_h M_m}{K_{cf}}s^4 + \frac{B_m M_h}{K_{cf}}s^3 + \left(M_h + M_m + \frac{K_{cp}M_m}{K_{cf}}\right)s^2 + \left(B_m + \frac{K_{cp}B_m}{K_{cf}}\right)s + K_{cp}}$$

A drawback of the cascaded (haptic)mass-compliance-mass-compliance design is that the system becomes of a higher order (in contrast to a second order mass-spring system), which greatly increases complexity. However, by a clever choice of module parameters, the dominant behavior of this system can be approximated by that of a second order system. Because the master virtual object should accurately track the operator position, and likewise the virtual object should closely track the telerobot, the configuration springs should be tuned as stiff as possible (see Figure 5.1). The coupling spring on the other hand, should provide stable and smooth coupling between the master and slave control objects. Therefore, from a mathematical point of view, choosing K_{cf} sufficiently large ($K_{cf} \gg K_{cp}$) and $M_h \ll M_m$ the 4th order equation reduces to a 2nd order system with transfer function:

$$H(s) = \frac{X(s)}{F(s)} = \frac{1}{M_m s^2 + B_m s + K_{cf}} \tag{5.1}$$

Intuitively this means that preferably $K_{cf} \to \infty$ and $M_m \to 0$ (note that this is in accordance with the notion of perfect telemanipulation), and these parameter values could in theory be chosen as such in the ideal situation. However, due to discretization of the system, the parameter settings are restricted by the sampling frequency of the discrete subsystem. This makes that upper and lower limits exist for the stiffness and mass, which relate to the natural frequency of the controller. This is not a drawback of the architecture, since in theory any sampling frequency can be chosen for an application, but rather a limitation imposed by, and inherent to, the design of the actual (teleoperation) system. For the youBot, the sampling frequency is $f_s = 1000\ Hz$, which means that the allowed bandwidth of signals in the system is limited to a maximum $500\ Hz$ (by the Nyquist criterion; $f_{max} \leq 0.5 \cdot f_s$). As such, the highest natural frequency ω_n arising from the IPC modules must always remain below half the sampling frequency. Certainly, this poses a strict limit on the achievable transparency.

Taking $\omega_{n,max} = 500$ and a haptic device mass $M_h = 45\ g$, the values for the master object and configuration spring can be determined. Using $M_m \ll M_h = 0.01\ M_h = 0.00045\ kg$ the value for K_{cf} becomes:

$$\omega_n = \sqrt{\frac{K}{M}} \tag{5.2}$$

$$500^2 = \frac{K}{0.00045} \quad \rightarrow \quad K_{cf} = 112.5\ N/m$$

Then, in order to approximate the second order system, taking $K_{cp} \ll K_{cf} = 0.01\ K_{cf}$ results in a coupling stiffness to the communication channel of $K_{cf} = 1.125\ N/m$, which is far too low to guarantee good transparency (see e.g. experimental values from Table E.3). To that end, since the hand mass of the human operator will also add to the haptic device mass during operation, it is chosen that $K_{cf} > K_{cp} = 10\ K_{cp}$ and $M_m < M_h = 0.1\ M_h$and as such:

$$500^2 = \frac{K}{0.0045} \quad \rightarrow \quad K_{cf} = 1125\ N/m$$

Hence, the values used for the spatial compliances are determined as $K_{cf} = 1125\ N/m$ and $K_{cp} = 100\ N/m$ for providing a moderate amount of coupling between the master object and communications. Accordingly, $M_m = 0.045\ kg$ yields an undamped natural frequency of $\omega_n \approx 500\ Hz$. The values for the moment of inertia and rotational stiffness have been scaled accordingly; for a spherical object with $j = 0.4\ m\sqrt{r}$. With some damping, the system resonance frequency $\omega_d = \omega_n\sqrt{1-\zeta^2}$ will be somewhat lowered, providing some headroom. Furthermore, the overshoot is determined by the amount of damping, which is commonly chosen about 0.7 times the critical damping. This results, taking $\zeta = 0.7$, in an approximate resonance frequency $\omega_d = \omega_n\sqrt{1-0.7^2} \approx 0.7\omega_n$. Hence, using:

$$\zeta = \frac{\mathrm{B}}{\mathrm{B}_{\mathrm{critical}}}$$

and

$$B_{critical} = 2M\omega_n$$

from which it follows that $B = 0.7 \cdot 2M\omega_n = 3.15$.

These variables are tunable to best suit an operators' wishes, but as a first indication the values calculated are applied to the system from Figure 5.2. A force is applied to the haptic device:

$$F = 5 \cdot [\, 0;\ 0;\ 0;\ \sin(time)\, ;\ 1 - \cos(time)\, ;\ 0\,]$$

which results in a circular motion of the haptic device and master virtual object, starting at and returning to the origin. This yields the haptic device and master object motions fromFigure 5.3. Note that due to the limited difference between K_{cf} and K_{cp} a noticeable tracking error occurs between haptic device and master object, which could be reduced by increasing the factor between the coupling and configuration stiffness.

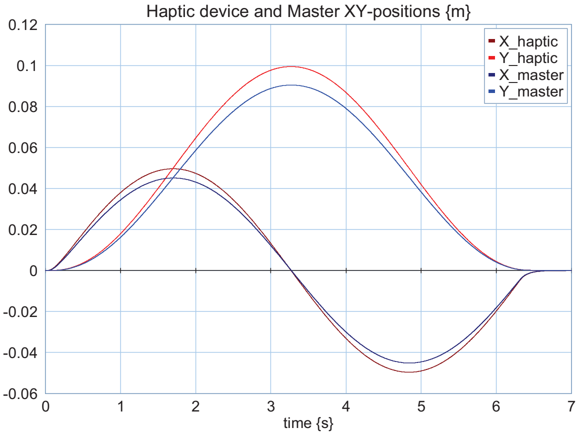


Figure 5.3: Master object tracking trajectory

5.3 Wave variable control

The next step in the simulation considers implementation of the wave modules, using both master and slave controllers. Before addressing the complete teleoperation system the behavior of the IPC master and slave objects will be addressed when connected to a delayed transmission line by coupling springs (note that this represents the dominant second order system behavior from section 5.2.1). The system under consideration is presented in Figure 5.4. Here, the Z-blocks represent the wave modules as presented in section 4.3. The ZOH (zero-order-hold) and sampling blocks are included since in reality a transmission line is a physical medium and therefore continuous in nature. The communications block (COMM) represents the line and associated delays.

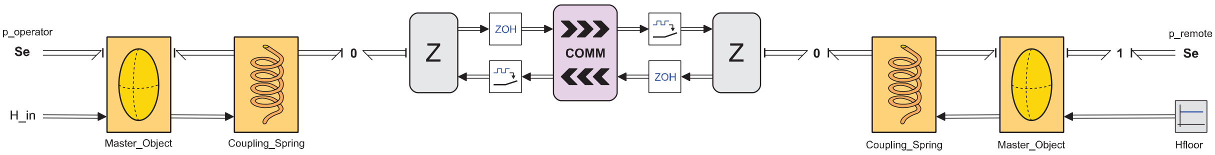


Figure 5.4: Master and slave objects connected by delayed transmission line

Using the same parameter settings for the controllers calculated in section 5.2.1, results in some critical issues and overall system instability. Because the master and slave springs are connected on both sides of the channel and are of (approximately) the same stiffness, the springs start oscillating across the communication line. Especially in case of small delays the communication line adds a considerable amount of delay to the signal travelling the line, which basically induces as a small amount of phase lag. A packet of information then starts rebounding across the communication line. This effect is clearly visible from Figure 5.5, where in the encircled graph it is visible that this packet bounces between the master and slave sites, in this case made visible by the effect on the X-position of both virtual objects. The frequency at which this packet rebounds over the line can be called the 'natural frequency' of the communication line (with the delay of the line at $T = 100\ ms$, the roundtrip delay of a signal completely crossing the line and back is $200\ ms$, i.e. $\omega_{cl} = 5\ Hz$). Note that this effect is not caused by an impedance mismatch, but rather due to the stiffness elements on either side of the channel, which cause the packet to 'bounce' from one side to the other and back.

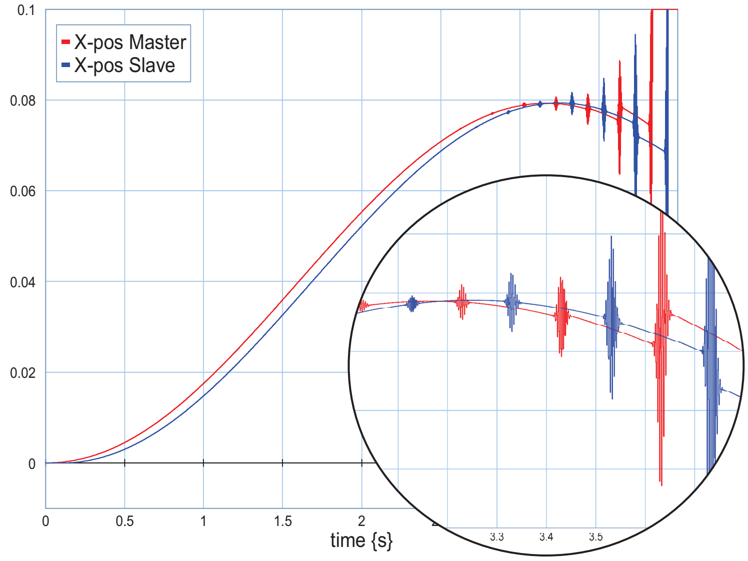


Figure 5.5: Effect of packet rebounding across the communication line

5.3.1 Wave filtering

A quick solution to the issue just addressed would be to lower the gain of the feedback loop, though this induces a tracking error between the master and slave virtual objects. Alternatively the mass and stiffness parameters could be readjusted to a more conservative value, though this clearly is undesirable with regard to achievable system performance. Because the issue is triggered inside the feedback loop of the communication line, a more attractive solution would be to use a wave filter inside the loop. As introduced in section 4.3.2, by inserting this first order filter in the wave feedback path, the loop can be filtered without affecting the direct signal paths that carry the velocity and feedback force. This way, by filtering at the relevant natural frequencies, system stability is retained while at the same time preserving the velocity and force signals. Figure 5.6 shows the system response for a half-circle motion from the origin to $[\,0,\ 0.25,\ 0\,]$ for a variety of transmission delays. The wave filter in this experiment has been tuned with $k = 1$ and $\tau = 0{,}1$ which puts the crossover frequency at $10\ Hz$ as substantiated by the discussion in section 4.3.2.

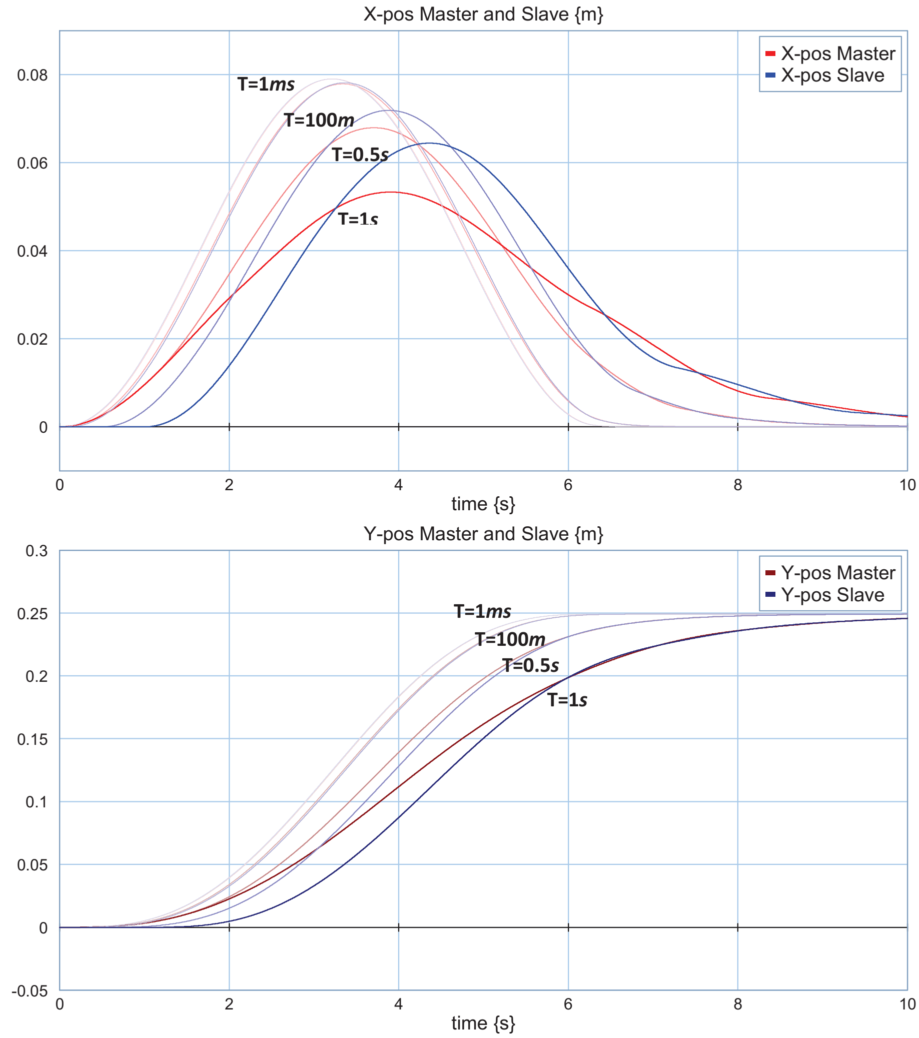


Figure 5.6: Master and slave object X and Y-positions for $T_{delay} = 1, 100, 500$ and 1000 ms

As the delay increases, the master and slave positions show a larger tracking error, which clearly is to be expected. However, the system shows smooth behavior for both the velocity and position signals, and near perfect convergence of both virtual objects (steady state error $\ll 1\ mm$ for the final Y-axis position). It should be noted that placing the wave filter only in the direct feedback path of the S^{+} wave (see Figure 3.4) the feedback loop inside the communication line is filtered, there is no filtering in the direct forward and return paths. Consequently, the suggested wave filter will only prevent instability induced in the inner loop of the communications and will not provide damping for signal loops that exist outside the communication channel.

5.4 The complete teleoperation scheme

Before testing the controller design on the youBot model, the behavior of the overall bilateral teleoperation scheme is considered, and the effect of tuning the scattering matrix $\boldsymbol{N}$ is verified. The complete scheme is shown in Figure 5.8. consisting of the operator, master and slave IPC control, the communication line, and the remote robot with which the operator interacts. First, a virtual object with mass $M_r = 2.5\,kg$ is used to represent the remote robot and evaluate the effect of tuning the scattering impedance matrix on the response of the system, using a constant time delays of $T_d = 0.5\,sec$. The user applies a force to the haptic device as shown in Figure 5.7.

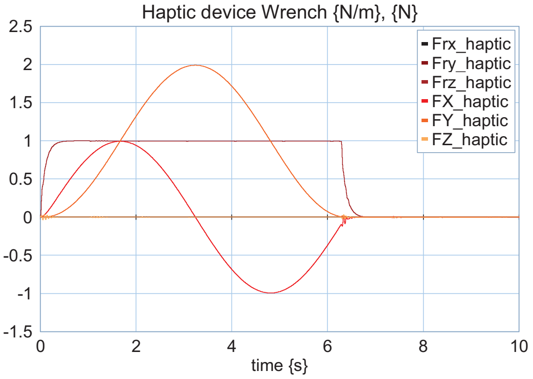


Figure 5.7: Force profile from haptic device

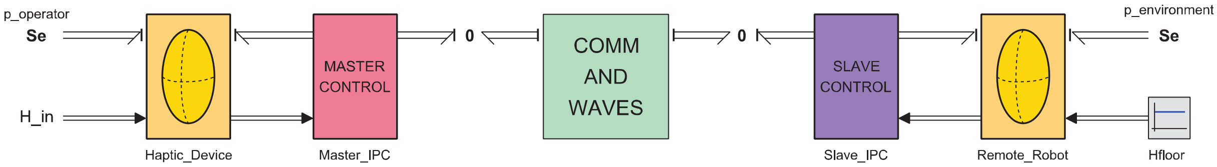


Figure 5.8: Complete teleoperation scheme with IPC controllers and wave variables

Figure 5.9 shows the result for several values for the scattering impedance $\boldsymbol{N}^{-1} = 0.4I,\ 0.2I$ and $0.1I$ and how this affects system response. The plots on the left show the X and Y motion of robot and haptic device, and the plot on the right shows the position error between master and slave (E_HS; dotted line) and between haptic device and robot (E_HR; solid line). As the damping of the wave transformation increases, the response of the master becomes slower but the tracking accuracy improves. Since a smaller time delay is also beneficial for the tracking accuracy, this means that for larger delays a higher scattering impedance is preferable. However, though using a larger damping with increasing delay improves tracking, the settling time of the system grows exponentially, which clearly is undesirable. Furthermore, it is interesting to notice that without the wave filter (proposed and verified in the previous section) the system becomes unstable when further running the simulation for $\sim 60\ sec$ with no input force applied. Close inspection reveals a packet of information rebounds over the communication line, being amplified with each roundtrip. Note the encircled graph which shows a magnification of the signal convergence around time $t = 25\ sec$, clearly showing an offset for all simulation runs. Since the vertical axis scale is read in meters, a steady state error ranging up to $e_{ss} = \sim 2.5\ mm$ is visible, which is quite much for precision applications. Moreover, this error is cumulative and might therefore become considerably large during longer runs. The next section considers methods to improve tracking performance.

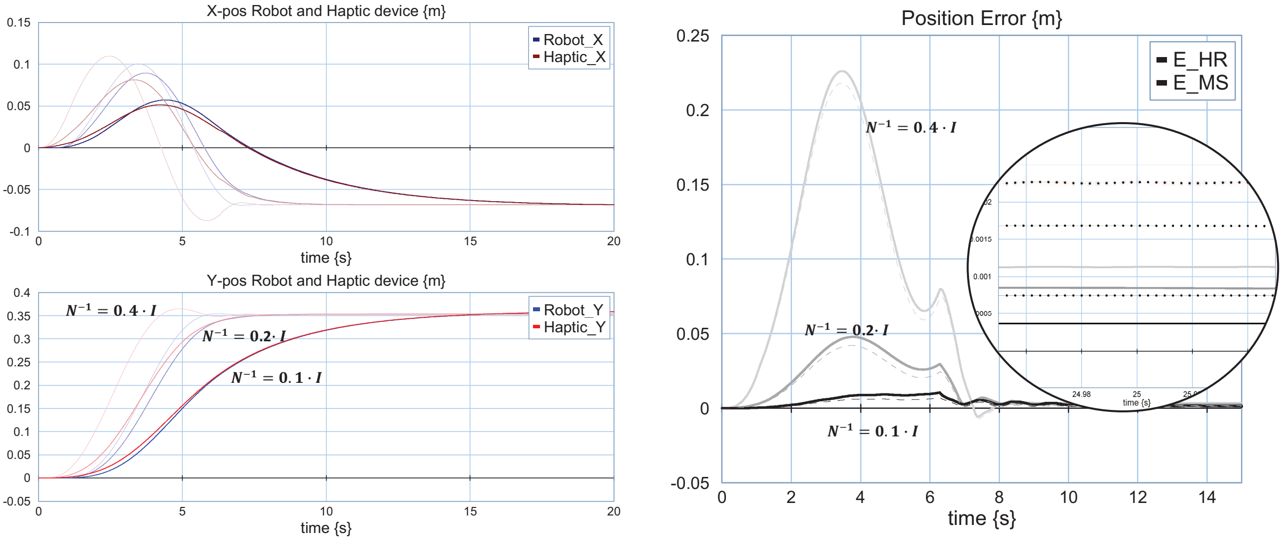


Figure 5.9: Lowering the scattering impedance improves tracking at the cost of settling time

5.4.1 Tracking & drift control

As was already addressed in section 3.4.3 and indeed noticed in simulation in the previous section, the use of wave variables is prone to considerable tracking errors, certainly if delays become large or long simulations are run. Though this can be partly solved by increasing damping, this adversely affects settling time of the system. To that end, additional control methods for improving tracking and minimizing steady state errors, are essential to a design aiming to be reusable. Because the homogeneous position information is available throughout the system, a straightforward approach is to apply position feed forward control to the slave, such that its virtual object position becomes directly dependent on the master object (recall BP-scheme; from section 3.4.3). The resulting scheme is depicted in Figure 5.10. By feeding the haptic device – robot position difference directly to the spatial compliance connecting the slave object to the channel, the rest length of the coupling spring can be modified to match the remaining position error present between the master and slave object. Alternatively, the wave integrals could be sent across the communication line providing direct position information in a similar way. This may be a useful approach in situations where available communication bandwidth is limited.

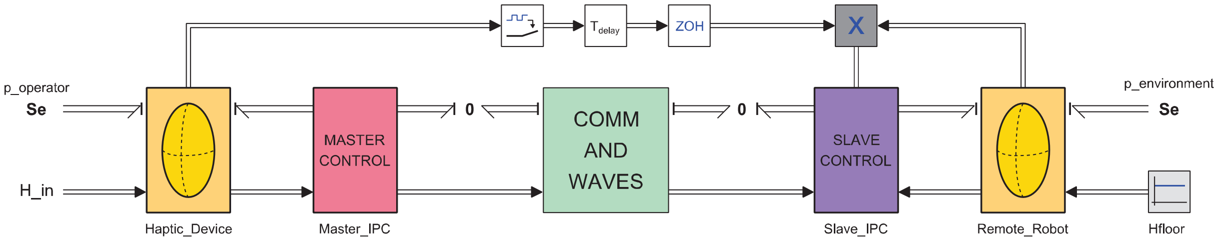


Figure 5.10: Teleoperation scheme with position feed forward

5.4.2 Varying delays

So far only constant delays have been considered. Therefore, the next step is to investigate the effect of varying delays on the design, by introducing a stochastically varying delay into the communication line. Using the original scheme as presented in Figure 5.8, the system is highly susceptible for varying delays and will show a dramatic increase in tracking error. The delay signal used is $T_{delay} = 0.3 + 0.25 \cdot S_{stoch}$, where S is a random stochastic signal between -1 and 1 at a bandwidth of $500\ Hz$. This results in a delay of approximately $0.05{\sim}0.55\ sec$, which seems reasonable with reference to the values used in Table E.1. The system behavior in case of the scheme from Figure 5.8 and scheme with tracking control are shown in Figure 5.11, where $\boldsymbol{N}^{-1} = 0.2 \cdot I$ and the operator commands the same force profile as before (see Figure 5.7).

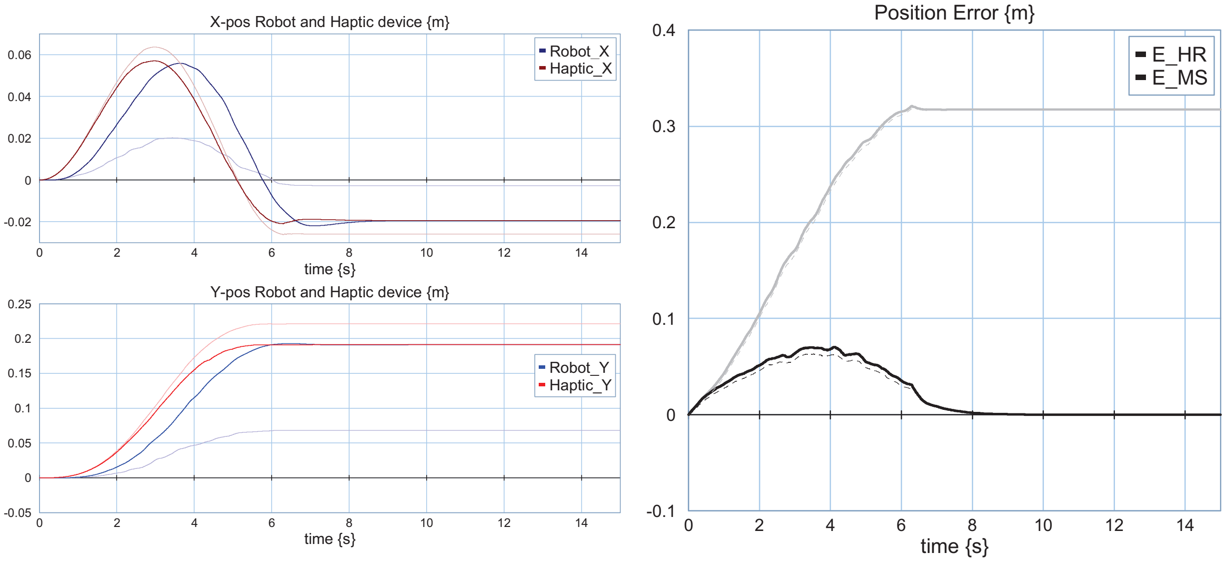


Figure 5.11: Teleoperation scheme without and with tracking control

From the graphs it is clear that the varying delay introduces loss of information in the system, resulting in very poor tracking performance. When using the setup from the section 5.4.1 though, the position of the master is sent to the slave side and is used to achieve absolute position feedback to the master, yielding greatly enhanced tracking performance and convergence of the master and slave, see the right plot of Figure 5.11.

5.5 Use-case; the youBot

The main goal of this research is to create and easily implementable, reusable architecture, which is able to provide at least basic functionality and acceptable performance to a wide variety of teleoperation systems. *Basic functionality* in this case refers to providing a complete controller platform for providing an operator with the tools to bilaterally operate a remote robot over a delayed transmission line. *Acceptable performance* (though partly application specific) has been defined by the criteria and considerations outlined in section 3.2, which have served as a guideline. In order to verify whether the proposed architecture is indeed capable of meeting these requirements, the design is tested on the youBot arm and base.

5.5.1 The omnidirectional base

In this section, control of the youBot base using the proposed architecture is analyzed, see Figure 5.12. The complete controller design is used, with the proposed wave filter and tracking scheme.

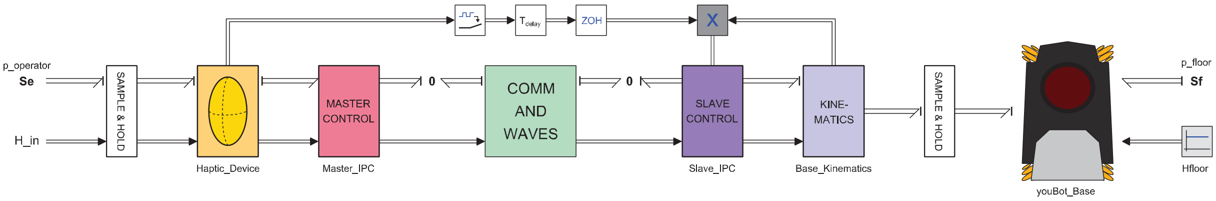


Figure 5.12: Architecture for the youBot base teleoperation system

As has been confirmed throughout this chapter, there are several ways of adapting the dynamical behavior of the system. The three main components which can be tuned in order to improve system behavior are the impedance matrix, adding a corrective term to change the coupling springs' lengths, and changing the IPC parameters for the virtual objects and springs. Additionally, the wave filter as introduced in section 4.3.2 has been employed to provide high frequency damping in the position path and to ensure stability of the inner communication loop. All of these control strategies have been used with the youBot base and the results are presented here.

In this experiment the operator commands the platform to move in a square by applying a force to the haptic device as shown in Figure 5.13. The communication channel has been modeled in the same way as in section 5.4.2, i.e. a stochastically varying delay. The associated motion of the youBot base can be found in Figure 5.14.a, which represents the floor plane, with on the horizontal bar the X-position and on the vertical bar the Y-position. The effect of tuning the impedance matrix is made visible for $N^{-1} = 0.2 \cdot I$ for the bright lines, and $N^{-1} = 0.1 \cdot I$ for the pale lines. The error between the haptic device and base model (E_HR) and between position of master and slave virtual objects (E_MS) are shown in Figure 5.14.b for the same values of the impedance matrix.

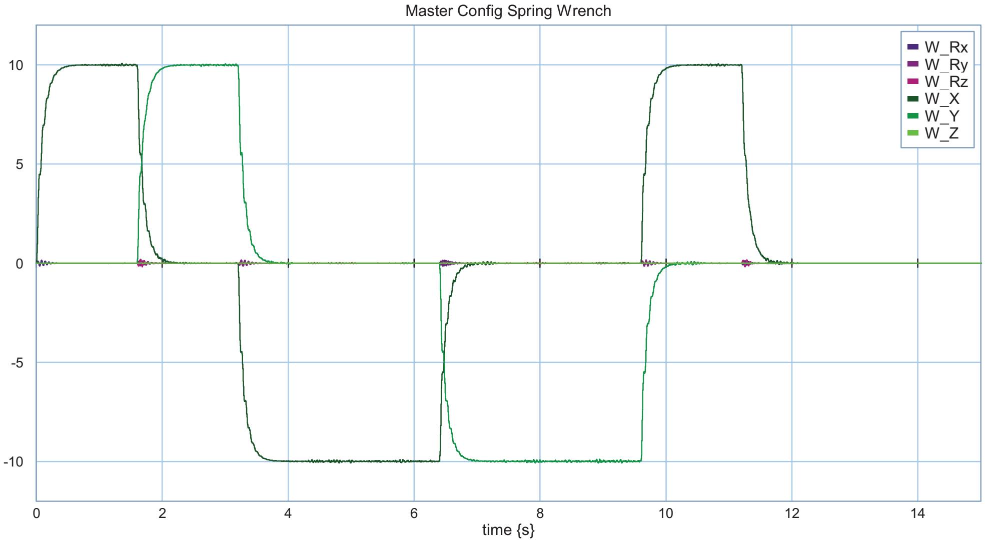
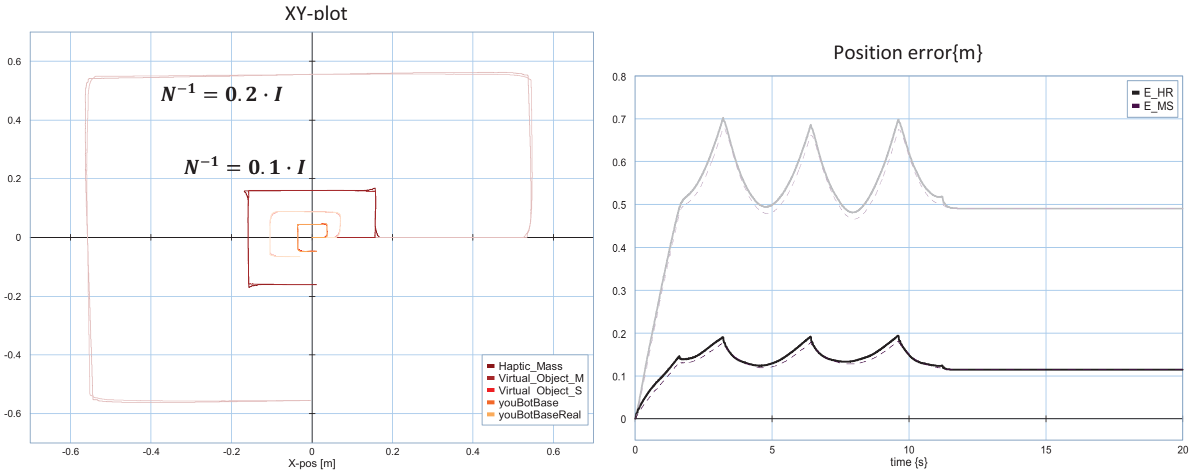


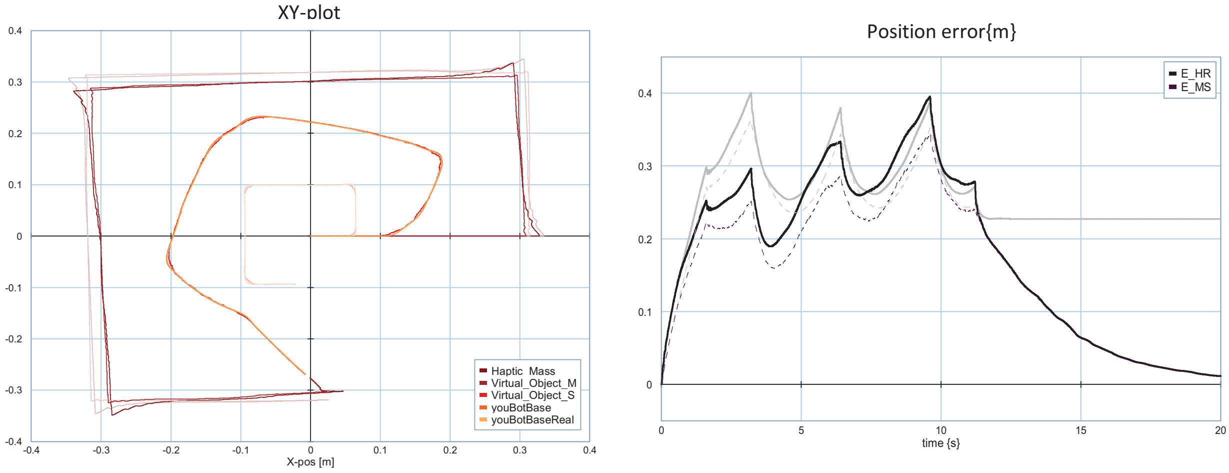
Figure 5.13: Force exerted by master coupling spring (connecting haptic device to master virtual object).



a. Trajectory of the base b. Position error for E_HR and E_MS

Figure 5.14: Control of youBot base with only scattering variable control.

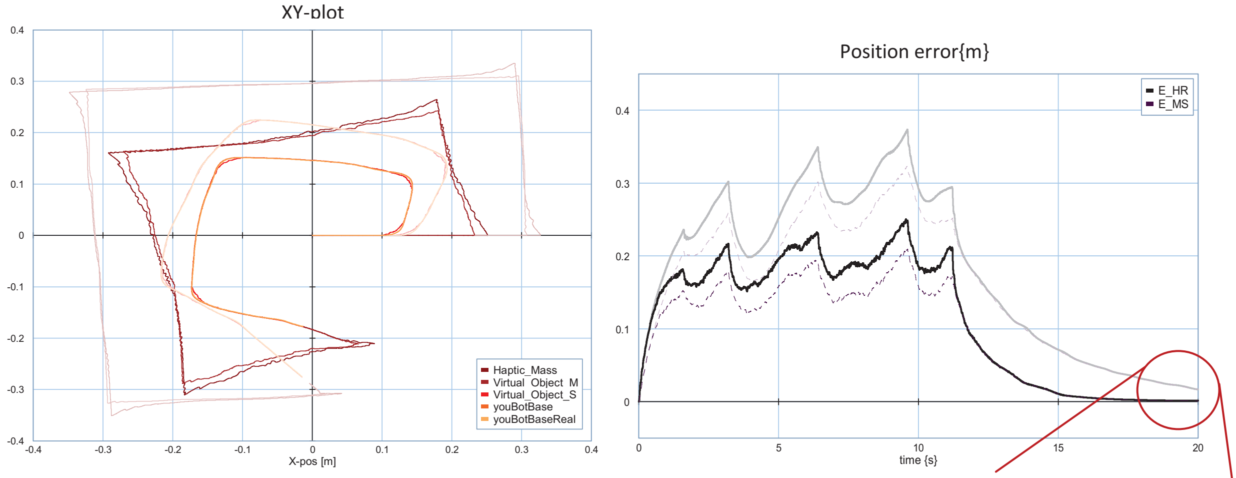
As expected, the effect of increasing the impedance of the communication channel results in a slower motion, but as a logical consequence, also a smaller error. Due to the varying delays and loss of information however, it can be seen from the position error that though the error decreases it does not converge. Using the proposed position feedback for varying the rest length of the controller springs yields, in case of using the master virtual object position and applying this to the slave coupling spring, the behavior depicted Figure 5.15 a and b. The pale lines represent the situation without position control, while the bright lines are acquired when enabling position control. Note that for this experiment the input force has been increased in order to obtain a meaningful movement of the base. This also results in a larger position error with regard to the simulation results from Figure 5.14, but this is to be expected for bigger movements (i.e. the relative error remains approximately the same). From the first figure it is clear that the slave object tends more toward the master object, which corresponds to a better tracking behavior and smaller position error, as confirmed by the position error plot.



a. Trajectory of the base b. Position error for E_HR and E_MS

Figure 5.15: Control of youBot base when adding master position feedforward.

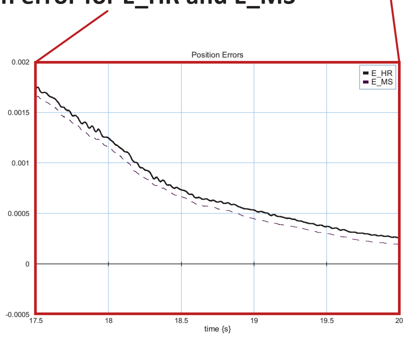
Instead of using a one-directional position transmission over the communication channel one could also, as motivated in sections 2.4.4 and 3.4.3, choose to communicate both master and slave virtual object positions and use these to modify the rest length of the coupling springs at both sides of the channel (i.e. in both the master and slave controllers). In that case, in addition to the slave object being ‘pulled’ toward the master side, also the master object will be ‘pulled’ toward the slave object through absolute position feedback. The result of applying this control strategy to the model results in improved behavior as shown in Figure 5.16.a and b. In this figure, the pale lines represent the simulation with only position control to the master, and the bright lines represent the situation in which both sites receive absolute position information.



a. Trajectory of the base

b. Position error for E_HR and E_MS

Figure 5.16.a : Control of youBot base when adding slave position feedback.



From the magnified portion of the graph it can be seen that at time $t = 20\ sec$ (or approximately $8\ sec$ after the last control signal from the operator) the position error has decreased to within $0.5\ mm$. Because the youBot base is a relatively heavy object ($\sim 20kg$), convergence speed and consequently also drift behavior may be expected to improve for lighter telerobots. Additionally, the value of scattering matrix could be increased, though this will result in slower movement of the robot.

5.5.2 The 5-DOF arm

The last section of this chapter covers the verification of the proposed control method when used on the youBot arm. The complete schematic used in this simulation can be found in Figure 5.17.

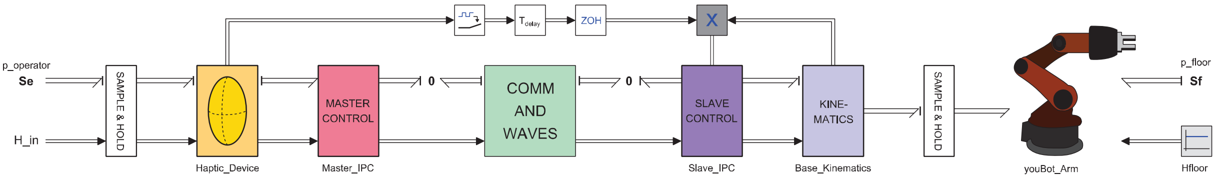
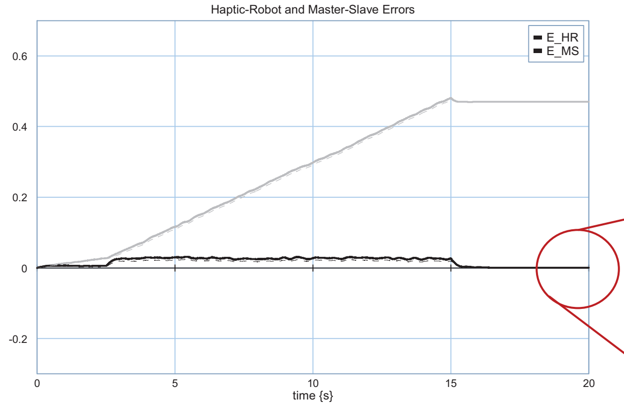


Figure 5.17: Architecture for the youBot arm teleoperation system

In this experiment, the operator commands the arm to move from its original pose (Figure G.8.a) to an operational pose (Figure G.8.b). First, the simulation is run without additional position control and next with position control from both the master and slave side. A similar result as with the base is obtained for the tracking experiment, see Figure 5.18, where the tracking error reduces to within $1mm$ approximately 3 seconds after the last input is applied by the operator. Additional results as well as the motion performed with the arm can be found in Appendix G.

From the figure it can be seen that with the position control active a small position error is maintained during the entire length of the motion, and dropping quickly after the operator stops interacting. From the magnified window it can be concluded that the error drop to about $0.4mm$ at time $t \approx 17\ sec$.



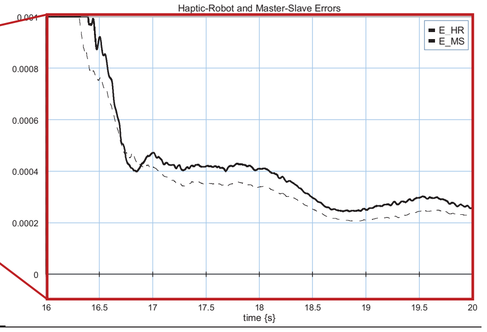


Figure 5.18: Tracking error without and with position control

For the second experiment, a test of the youBot arm when making contact with a wall has been tested. With the arm in the operational position from the previous experiment, the arm is commanded to move in the Z-direction, with an obstruction in the form of a solid wall, see Figure 5.19. The wall is represented by similar values being used for the youBot platform contacting the floor; a damping element with $R = 1000\ [Ns/m]$ and a stiffness element with $K_{wall} = 1 \cdot 10^7\ [N/m]$, connected to a zero-flow source. The MTF element calculates the point at which contact occurs based on the youBot arm end-effector.

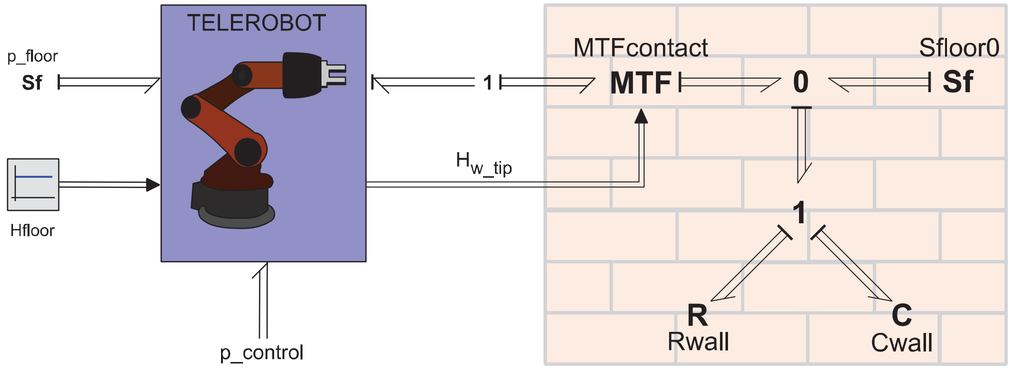


Figure 5.19: youBot arm in contact with wall

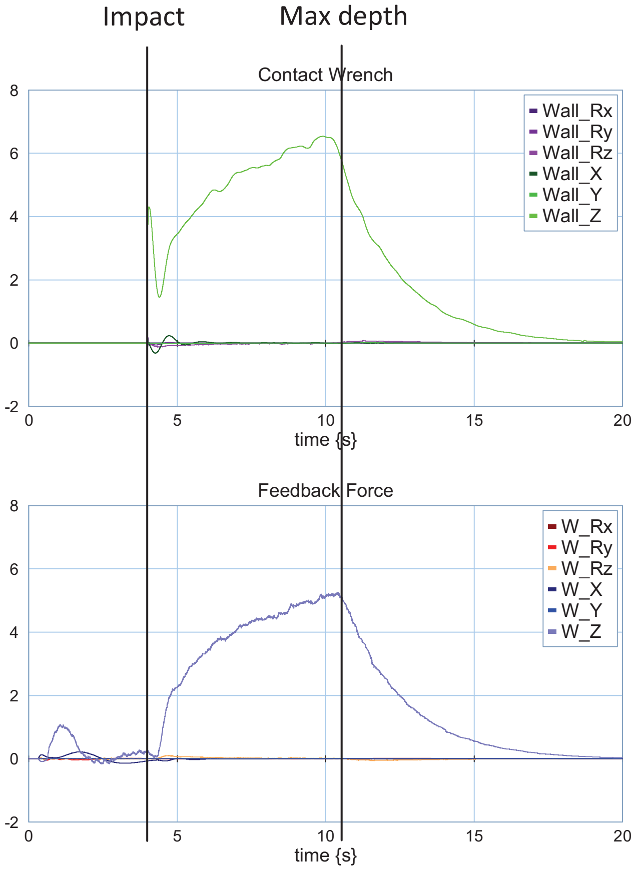


Figure 5.20: Contact force at youBot arm tip (top) and feedback force to user (bottom)

The results for this experiment are displayed in Figure 5.20. Contact with the wall at the slave side takes place at $t \approx 3.7\ sec$ and hence at the side of the operator is felt approximately $300\ ms$ later. This because the transmission line in this experiment is again modeled as a varying delay of the form $T_{delay} = 0.3 + 0.25 \cdot S_{stoch}$ (recall section 5.4.2). Notice that the initial bump with the wall is not noticed by the operator, and that there is a difference of approximately $1\ N$ between the actual force and the sensed force, but overall the profile shows good reflection properties. The two lines in the figure mark the point of impact with the actual wall and the maximum 'penetration depth' by the master virtual object. This can also be seen from Figure 5.21, which shows the motion profiles of the robot, slave, master and haptic device (Hw_R, Hw_S, Hw_M and Hw_H) in the direction perpendicular to the wall. The extension of the spatial springs and the coming into contact with the wall of the youBot arm is also clearly visible. Stable contact was also achieved with several other materials with lower stiffnesses and less damping.

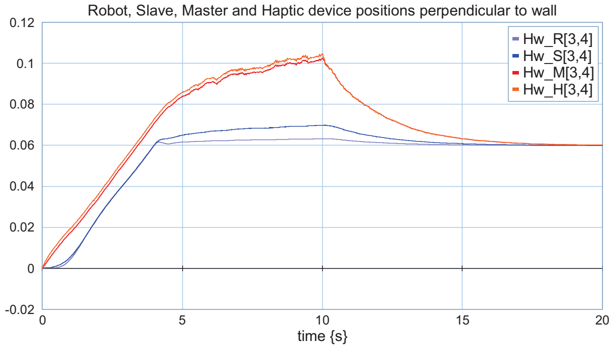


Figure 5.21: Motion in Z-direction for the robot, slave, master and haptic device (right)

5.6 Results

In this chapter the proposed control framework was analyzed on several aspects and was finally tested on a complete telemanipulation setup (the youBot platform and arm) when being connected to a communication line with nonnegligible time delay. Using the wave modules to map power variables to wave variables during transmission over the communication line and intrinsically passive control (IPC) for guaranteeing good contact, several extensions were applied to improve behavior of the overall system.

5.6.1 Verification of criteria

This section discusses the results of the generic and use-case specific simulations performed throughout this chapter, with respect to the criteria covered in section 3.2. The most important goal for the control framework was to guarantee stability under all circumstances. Though IPC and the wave variable method are passive by design and were therefore selected for use in the architecture, it can be concluded that system stability can still be compromised. There are two main aspects to consider with regard to this matter. First, as the experiment in section 5.3 shows, and was also underlined in section 4.3.2, instability of can occur due to the rebounding of information across the communication line. Besides impedance mismatching, for which solutions are present (section 2.3.3), it can still occur that signals or 'pieces of information' circle the inner communication line loop (recall Figure 2.5). An example of this is due to the IPC stiffness modules as was noted in section 5.3. A solution to this issue was evaluated in section 5.3.1. Secondly, stability can be compromised due to bad tuning of the IPC modules. The cause of this is due to the digital nature of the system; if too high (natural) frequencies of the system arise with respect to the sampling frequency, signals change too rapidly and the system can become unstable. This is however not a drawback of the control framework but rather something inherent to any digital system, requiring for a careful design and running of system parameters. Additional control in the form of smart energy bookkeeping can be used to cope with these issues, which is well addressed in literature, see e.g. (Secchi et al., 2003).

With respect to the dynamic properties which can be used to comment on the transparency (refer to Table 5.1) of the overall system, the achievable performance is primarily limited again due to the sampling frequency of the system. As was addressed in section 5.2, tuning of the spatial compliances and virtual masses should be chosen as high as possible, but in such a fashion that natural frequencies arising in the system remain well below (half) the sampling frequency. In this particular setup with the youBot, the IPC could not be tuned completely in such a way that a dominant 2nd order system behavior is achieved. Hence the virtual objects contribute to a weight of approximately $0.045\ kg$ and a maximum obtainable stiffness of $1125\ N/m$. In order to provide enough damping, the virtual objects were equipped with dampers and also the wave modules incorporate damping. The most important aspect with respect to transparency is that tuning of the control parameters is an application specific task.

Lastly, as noted earlier, the wave variable method does not excel in accuracy. Certainly with variable delays, the tracking error and drift become disproportionally large and the operator will feel completely disconnected from the remote robot. In order to correct for these errors, several methods of transmitting position information of the master and slave were addressed. This information can be used to provide feedback to the spring coupled to the communication line in order to compensate for the drift. This way, the steady state error has been significantly lowered in the simulations that were performed. Tracking errors can be minimized by increasing the scattering impedance, though this will also increase task completion time. Since direct position feedback is used these results are expected to hold even for tests that are run for a larger duration. The only errors that may accumulate during operation are due to discretization effects, but, as also noted in (Tanner and Niemeyer, 2004), with computing power nowadays constantly increasing, this effect will become less and less significant.

5.6.2 Reusability and modularity

Reusability of the modules has been verified by using the youBot base and arm as two separate use-cases. With regard to Figure 4.1, all behavior specific elements were used in simulations with both the base and arm. Throughout the experiments the only parts which have been edited are the parameter settings for both the IPC and wave variable modules; i.e. there have been made no structural changes. Wave filters and position control have been employed where necessary, and also for these elements only the parameter settings have been changed. It should be noted though, that for specific applications it may be necessary to relocate or improve the wave filters, or select a different method for providing absolute position control by using any of the three methods provided in section 3.4.3 (the WI, WC or BP-scheme).

6 Evaluation & conclusion

In the final chapter of this report, concluding remarks based on the results obtained during the course of this project and recommendations for future work to extend on what has been done, are provided.

6.1 Conclusion

The main objective of this research has been to construct a reusable controller design, for use in bilateral teleoperation systems, expressed in the following goals, as mentioned in section 1.3:

- Evaluate existing control methods for teleoperation systems
- Design a generalized controller framework and interface strategy
- Modularize the controller architecture to provide reusable elements for similar applications
- Explore tuning of the control modules and the effect on overall system behavior
- Verify performance and reusability of the design by doing simulations with the youBot

Based on an analysis of control methods for bilateral teleoperation available in literature, several control methods for designing reusable modules have been selected. The basis for the architecture has been formed by passive control methods in order to be able to guarantee stability in the presence of delays and interaction. The proposed architecture may indeed provide a solid basis for a bilateral telemanipulation systems, as it provides the tools for coping with these issues. Modularity of the design is only achievable to a certain degree and the aim has been to incorporate just enough functionality in a module to allow for intuitive and basic behavior. Reusability of the blocks is possible, though the wave variable modules are particularly specific in their operation, in the sense their use fixes the way information is sent across a communication line. Also, using the wave variable method has been criticized for limited accuracy, which means that additional control is a necessity. The IPC modules on the other hand, allow for reuse tot a larger extent due to the fact that they are based on physical concepts, which arise everywhere modeling and control applications.

6.2 Recommendations

Since the experiments in this research have been solely based on simulations, an obvious next step would be to implement the controller on an actual setup of the youBot. It should be possible to directly implement the control modules (i.e. IPC, wave modules and additional control) in a design. However, because it has been assumed that position and velocity information is readily available from the youBot, it is recommended that the framework be first evaluate with the design elements being used with the physical setup. Also, though simulations were performed on two subsystems (the youBot arm and base) which have considerably different dynamic properties and kinematics, testing with additional applications might provide a broader evaluation for reusability of the modules. Additionally, the modules might be added to a library for robotic telemanipulation applications, in order to provide off-the-shell building blocks.

Regarding discretization of the signals and vice versa, the sample & hold elements have not been optimized for the application, due to the simulation nature of this assignment. However, these elements can be designed to track energy flows during the conversion from the digital to the continuous domain and vice versa, for the purpose of guaranteeing passivity. Several studies address this topic; see e.g. (Secchi et al., 2003).

With respect to comments made in chapter 1, the modules developed are not yet numerous enough to realize a ‘controller library’ in a concrete form. However, all models are available at the RaM institute at the University of Twente. Additionally, at this time the IPC and wave modules are essential for operation, which means that stability, in fact even normal operation, cannot be guaranteed upon removal of these modules.

Finally, since the wave variable method has been criticized for poor tracking properties and bad performance in the presence of varying delays, other methods based on energy monitoring have gained considerable interest. To that end, several studies have aimed at creating designs which monitor energy flows to ensure no energy generation is taking place anywhere in the system. Studies include (Franken et al., 2009), (Franken et al., 2011) and (Franken et al., 2011). The proposed framework, specifically the IPC modules, may also offer possibilities here. By writing an energy function for the virtual objects based on their relative positions, it should be possible to monitor master and slave energetic states (i.e. the states of the virtual objects). Because the master and slave virtual objects can be tuned to a designers’ wishes, and hence everything about the virtual objects is known, the maximum energy needed to bring the objects together can be exactly determined. As with the IPC used in this report, the spatial compliances can then be used to bring the local and remote objects together.

A Communication line passivity

Starting from Equation (2.4):

$$E_{in}(t) = \int_0^t \left(\dot{x}_m(\tau)\, F_m(\tau) - \dot{x}_s(\tau)\, F_s(\tau)\right)\, d\tau \qquad \textbf{(A.1)}$$

Assuming a constant time delay T, the wave outputs at the communication block can be written:

$$\begin{aligned} u_s(t) &= u_m(t-T) \\ v_m(t) &= v_s(t-T) \end{aligned} \qquad \textbf{(A.2)}$$

Rewriting the Equations (2.3) for $\dot{x}$ and F as follows

$$\dot{x}_i = \frac{1}{\sqrt{2b}}(u_i + v_i) \qquad \text{and} \qquad F_i = \sqrt{\frac{b}{2}}(u_i - v_i) \qquad i = m, s$$

and filling into Equation (A.1) yields:

$$E_{in}(t) = \frac{1}{2}\int_0^t \left(u_m{}^T(\tau)\, u_m(\tau) - v_m{}^T(\tau)\, v_m(\tau) + v_s{}^T(\tau)\, v_s(\tau) - u_s{}^T(\tau)\, u_s(\tau)\right)\, d\tau$$

Next, substituting the delayed wave outputs from Equation (A.2) gives (see also Figure A.1):

$$\begin{aligned} E_{in}(t) &= \frac{1}{2}\int_0^t \left(u_m{}^T(\tau)\, u_m(\tau) - u_m{}^T(\tau-T)\, u_m(\tau-T) + v_s{}^T(\tau)\, v_s(\tau) - v_s{}^T(\tau-T)\, v_s(\tau-T)\right)\, d\tau \\ &= \frac{1}{2}\int_0^t \left(u_m{}^T(\tau)\, u_m(\tau) + v_s{}^T(\tau)\, v_s(\tau)\right) - \frac{1}{2}\int_0^{t-T} \left(u_m{}^T(\tau)\, u_m(\tau) + v_s{}^T(\tau)\, v_s(\tau)\right) \\ &= \frac{1}{2}\int_{t-T}^t \left(u_m{}^T(\tau)\, u_m(\tau) + v_s{}^T(\tau)\, v_s(\tau)\right) \geq 0 \end{aligned}$$

representing the energy stored on the communication line. Because it always holds that $u_m{}^T(\tau)\, u_m(\tau) \geq 0$ and $v_s{}^T(\tau)\, v_s(\tau) \geq 0$ this means that energy can indeed only be temporarily stored on the communication line and thus not come from the line, hence obeying the passivity principle.

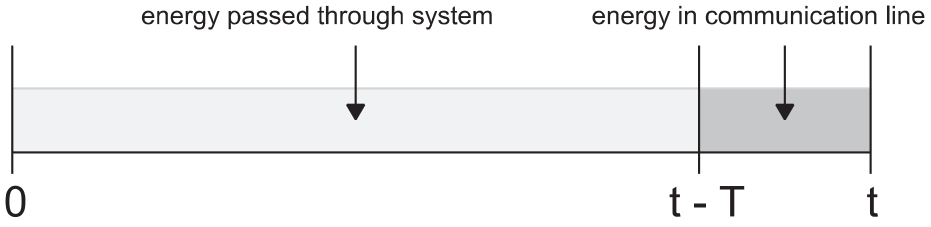


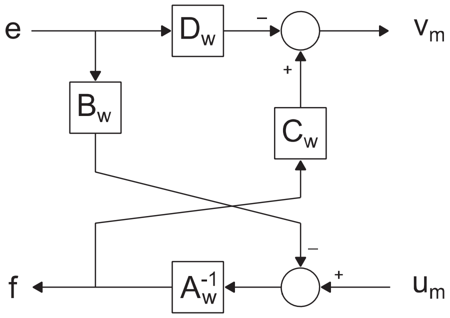
Figure A.1: Energy flow through wave coded communication line

B Notations

ψ_i	A right-handed orthonormal coordinate frame
H_i^j	Homogeneous coordinate transformation matrix from ψ_i to ψ_j
$Ad_{H_i^j}$	Adjoint map for a coordinate transformation of a twist from ψ_i to ψ_j
$T_i^{k,j}$	The twist of ψ_i with respect to ψ_j expressed in ψ_k
W_i^j	Wrench applied to a rigid body attached to frame ψ_i expressed in the coordinates of ψ_j
$\tilde{x}$	Tilde operator; maps a three dimensional vector to a matrix of the form $\begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$
I_i^j	Inertial tensor of a body attached to frame ψ_i expressed in the coordinates of ψ_j
P_i^j	Generalized momentum of a body attached to frame ψ_i expressed in the coordinates of ψ_j

C Schematic representation scaling matrices

The original schematic representation for the multi-DOF wave equations presented in chapter 2 can also be represented by the scheme in Figure C.2.a. This scheme can then be reorganized as shown in Figure C.2.b.



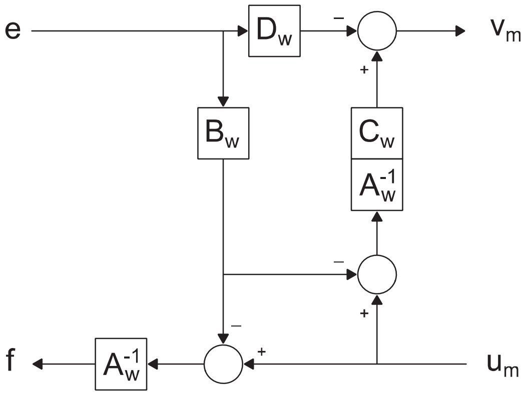
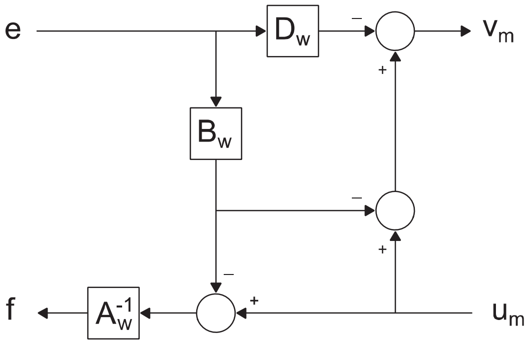


Figure C.2.a: Original multi-DOF scheme **b. Reorganized multi-DOF scheme**

Note that for the scaling matrices it holds that:

$A_w =$ Any nonsingular matrix
$B_w = (1/2)\ (I + S_w)\ A_w^{-T}$
$C_w = Q\ A_w$
$D_w = (1/2)\ Q\ (I - S_w)\ A_w^{-T}$

Consequently, by choosing for the matrices $Q = I$ (hence $A_w = C_w$) and $S_w = 0$ (hence $B_w = D_w$) will yield the schematic representations displayed in Figure C.3.a and Figure C.3.b respectively.



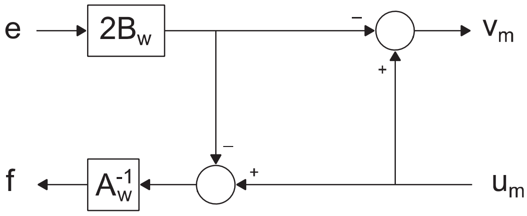


Figure C.3.a: Resulting scheme with $\boldsymbol{Q = I}$ **b. Resulting scheme with $\boldsymbol{Q = I}$ and $\boldsymbol{S_w = 0}$**

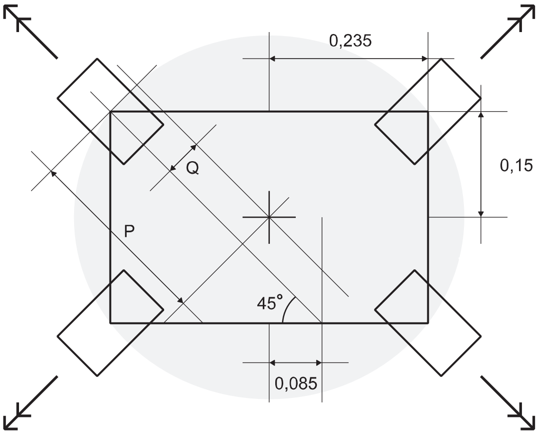
By setting furthermore $S_w = 0$ both matrices B_w and D_w become identical to each other and can be relocated to the effort and flows paths respectively. Since from the above equations it follows that for the chosen matrix values $B_w = \frac{1}{2} A_w^{-1}$, this means the left input (e, f) can be represented by a transformer element.

For neither of the schemes it is possible to merge the matrix transformations into two equal elements in the signals paths of $\boldsymbol{e}$ and $\boldsymbol{f}$, as is required for a $\mathbf{TF}$ element in bond graph modeling. The model from Figure C.3.b can only be dimensioned to fit the geometric scattering design by setting $B_w = \frac{1}{2} A_w^{-1}$ and then factoring the multiplier 2 as $\sqrt{2}$ in both the forward and return paths.

D Modified youBot base model

Specifications for the base:

The distance from the center of the base to any of the four wheel connections is $R = \sqrt{L_x^2 + L_y^2} = 0.279\ [m]$. The angle between a line running through the center of the base and a connecting point of a wheel and the X-axis is $\alpha = \tan^{-1}(L_y/L_x\ ) = 32.55\ ° = 0.568\ [rad]$.



It can be easily verified that $Q = \sqrt{\frac{1}{2} \cdot (0.085)^2} \approx 0.060\ [m]$ which represents the lateral distance between the direction of motion induced between a mecanum wheel and the floor, and the center of the youBot base frame. Applying a torque of $\tau = 1.0$ to the axes of each of the wheels will result (using a wheel radius $R_{wheel} = 0.0475$ [m]) in a total moment of

$$M = 4 \cdot \frac{\tau}{R_{wheel}} \cdot Q = 4 \cdot \frac{1.0}{0.0475} \cdot 0.060 = 5.1[Nm]$$

to the base. Additional properties of the base are a mass of $m = 20\ [kg]$ and a moment of inertia $I = 20\ [kg \cdot m^2]$.

The base kinematics model is constituted of the model displayed in Figure D.4 which is similar to the original youBot base model though with inverted causality. The "*Mecanum wheel*" submodels contain an altered version of the original submodel for a mecanum wheel, from (Dresscher et al., 2012).

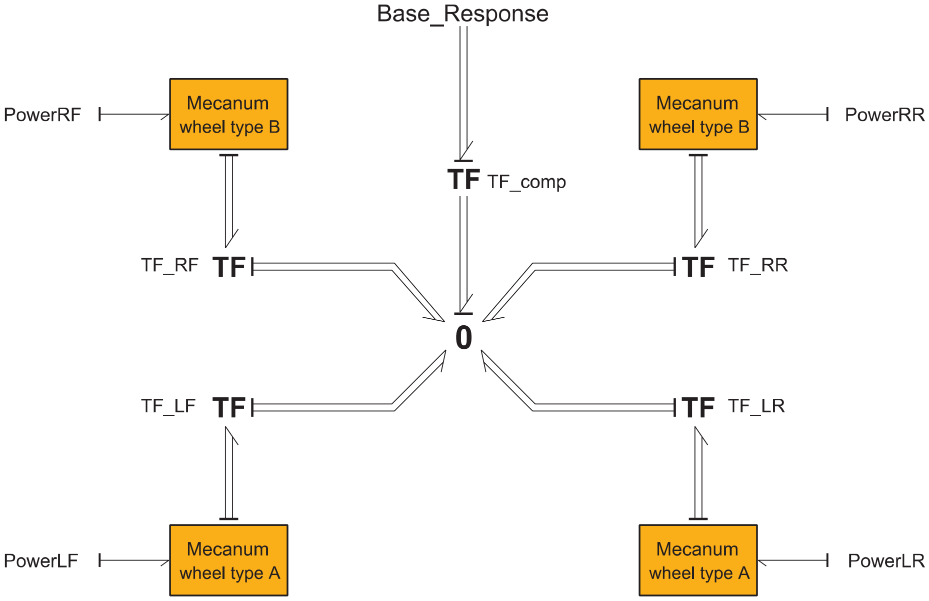


Figure D.4: youBot base kinematics model

The mecanum wheel submodels contain the model presented in Figure D.5. Here, the $\mathbf{TF}$: uTF element accepts a one dimensional power bond and maps this to the y-axis in a wrench vector. Since the element outputs a velocity, this means that forces acting in other direction than the y-axis do not affect motion of the base. The $\mathbf{TF}$: TFwheelRadius element maps the wheel axis rotation to an x-axis translation, i.e. to the contact surface of the wheel with the floor. In order to determine the direction of friction with the floor, which is the actual direction in which the wheel rolls, the $\mathbf{TF}$: TF_R element performs a 45° rotation around the z-axis. Next, the $\mathbf{TF}$: TF_roller discards all but the translational x component (the direction perpendicular to the roller) after which the $\mathbf{TF}$: TF_invR performs an inverse rotation of 45° rotation around the z-axis.

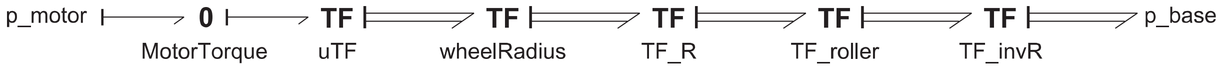


Figure D.5: Mecanum wheel submodel

The equations for the **TF** submodels are given as:

uTF

```
p2.f= [0; p1.f; 0; 0; 0; 0 ];
p1.e = p2.e[2];
```

wheelRadius

```
Hw = homogeneous( eye(3), [ 0; 0; -WheelRadius ] );
AdHw = Adjoint(Hw);
TrAdHw = transpose(AdHw);

p2.f = AdHw * p1.f;
p1.e = TrAdHw * p2.e;
```

TF_R

```
R = [  cos(-(pi/4)),      -sin(-(pi/4)),      0;
       sin(-(pi/4)),       cos(-(pi/4)),      0;
       0,                  0,                 1         ];

H = homogeneous( R, [ 0; 0; 0 ] );
AdR = Adjoint(H);
TrAdR = transpose(AdR);

p2.f = AdR * p1.f;
p1.e = TrAdR * p2.e;
```

TF_roller

```
AdR[1:3, 1:3] = 0;
AdR[4:6, 1:3] = 0;
AdR[1:3, 4:6] = 0;
AdR[4:6, 4:6] = [1, 0, 0; 0, 0, 0; 0, 0, 0];
TrAdR = transpose(AdR);

p2.f = AdR * p1.f;
p1.e = TrAdR * p2.e;
```

TF_invR

```
R = [  cos(pi/4),         -sin(pi/4),         0;
       sin(pi/4),          cos(pi/4),         0;
       0,                  0,                 1         ];
H = homogeneous( R, [ 0; 0; 0 ] );
AdR = Adjoint(H);
TrAdR = transpose(AdR);

p2.f = AdR * p1.f;
p1.e = TrAdR * p2.e;
```

It should be noted that the motion of the base model shows a discrepancy with regard to the original bond graph model for rotations about the Z-axis. This has been solved temporarily by scaling the Z-axis value by 6.7465 inside the *Base*-block. The resulting youBot base model calculates motion of the body frame based on axis-rotation and after inspection approximates motions of the original youBot base model closely.

E Internet-based control schemes

This appendix summarizes the results of an experimental comparison study regarding internet-based control schemes, as has been presented in (Rodríguez-Seda et al., 2006). Using four different internet models; LD-A, HD-A, LD-B and HD-B, for which the characteristics are listed in Table E.1, seven teleoperation control schemes were evaluated.

Measures	Experiments			
	LD-A	HD-A	LD-B	HD-B
Minimum Time Delay [ms]	68	468	48	448
Maximum Time Delay [ms]	96	496	144	544
Mean Time Delay [ms]	80	480	80	480
Standard Deviation of Time Delay [ms]	3	3	22	22
Packet Loss Rate [%]	10-15	10-15	45-55	45-55

Table E.1: Characteristics of the simulated internet models

The schemes used in the study are given in Table E.2.

Abbreviation	Control scheme
WS	Wave scattering transformation
DD	Digital data reconstructing filter
WI	Wave integral and reconstruction filter
P	Proportional control
PD	Proportional derivative control
WP	Wave prediction with energy regulation
AC	Passivity based adaptive control

Table E.2: Internet-based teleoperation control schemes

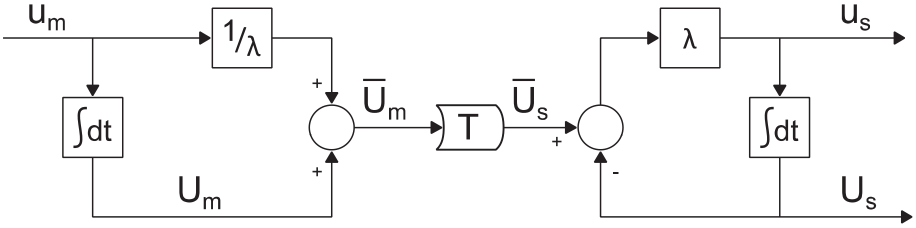
Each control scheme has been tested on five criteria; the transient error in free motion, the steady-state error both in free motion and constrained motion, the contact stiffness, and force-reflection error. Also, a measure for the complexity of the design has been evaluated by considering both the structural and computational complexity. Using a method called the Polyhedral Dynamics the structural complexity was determined, and the computational complexity by measuring the code length. In order to determine the best overall performance of each control method, the results for all four internet models has been summed for each of the control schemes. This has been done by manually determining the values from the original article (Rodríguez-Seda et al., 2006) for all schemes and all four internet models. Table E.3 presents these values, where all error measurements and the stiffness measure represent the total values obtained by summing the results of all four internet models. The total complexity is the sum of the structural (or static) complexity and computational complexity, the first being a measure for how well the components of the system are connected, the latter a measure for the amount of time and memory required by a computer to execute the control algorithm. The structural complexity is expressed as a value using Q-analysis, and the computational complexity is expressed as a normalized value in base of the shortest code. Also, for each scheme it is displayed whether the particular scheme uses the wave variable transform for transmitting information over the communication line. Finally, it is noted whether each scheme can theoretically guarantee stability under time-varying delays.

	WS	DD	WI	P	PD	WP	AC
Transient Error [rad] (free motion)	26.9	14.9	12.5	24.5	24.0	9.6	15.4
Steady-state Error [rad] (free motion)	2.58	1.85	0.50	0.73	0.90	0.50	0.78
Steady-state Error [rad] (constrained motion)	8.05	9.41	0.55	0.68	0.82	1.36	0.27
Contact stiffness [N/rad]	20.0	32.1	48.5	46.1	47.3	73.3	58.8
Force-reflection Error [N]	21.4	29.3	14.6	24.7	30.0	24.7	29.7
Complexity	3.97	5.09	5.26	3.67	3.69	6.27	4.77
Wave based	Y	Y	Y	N	N	Y	Y
Guarantees passivity (for varying delays)	N	Y	Y	N	N	Y	N

Table E.3: Combined result of all four experiment models per criterion

F Tracking & drift schemes

Wave Integral (WI) scheme; sending wave integrals in addition to the wave variables without affect passivity.



Integrated wave variable defined as (same for $V(t)$):

$$U(t) = \int_{0}^{t} u \, d\tau = \frac{bx + p}{\sqrt{2b}}$$

At the slave side, force is integrated into momentum:

$$p_s \int_{0}^{t} F_s \, d\tau$$

Incoming wave integral and momentum determine the desired position:

$$x_{sd} = \frac{\sqrt{2b}U_s - p_s}{b}$$

and finally, desired position and momentum specify returning wave integral:

$$V_{sd} = \frac{bx_{sd} - p_s}{\sqrt{2b}}$$

On the master side, the wave integral is computed directly from actual position and incoming wave integral:

$$U_m = \sqrt{2b}x_m - V_m$$

This extension implements following behavior:

$$x_{sd}(t) = 2x_m(t - T) - x_{sd}(t - 2T) - \frac{1}{b} \int_{t-2T}^{t} F_s(\tau) \, d\tau$$

which can also be computed directly, hence no need for calculating the momentum. The only requirement is to integrate F_s over a finite period of time. Combining wave signal with its integral during transmission:

$$\bar{U}_m = U_m + \frac{1}{\lambda} u_m$$

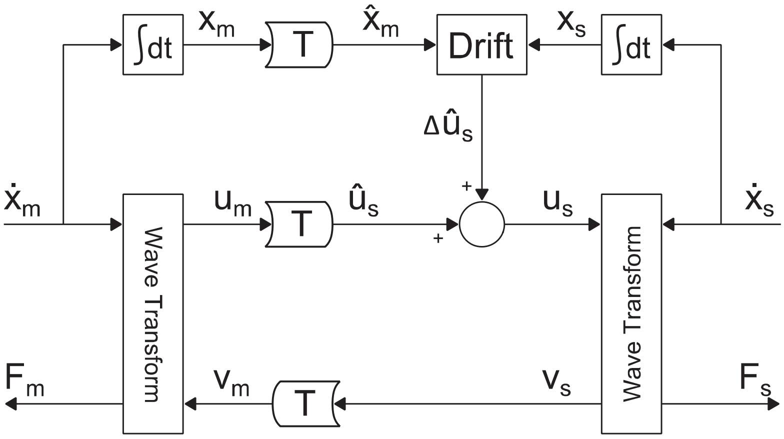
$$\bar{U}_s(t) = \bar{U}_m(t - T)$$

and separate via a stable first-order filter:

$$u_s = \lambda(\bar{U}_s - U_s) \qquad \text{and} \qquad U_s = \int_0^t u_s \, d\tau$$

where λ is the positive constant bandwidth of the filter. The choice of λ is arbitrary but must be significantly lower than the sampling rate of the digital implementation, i.e. $\lambda \ll f_s$. Effectively, $\bar{U}$ encodes all information required to operate the system; position, velocity and force.

Adjusting the Wave Command (WC); Drift control algorithm which can compensate for any source of steady-state error and variation in transmission delay by observing drift error between the two sides. The algorithm can also handle initial position offsets and temporary transmission losses.



Actual wave command u_m based on uncorrected value $\hat{u}_m$ and corrective term $\Delta\boldsymbol{u}_m$:

$$\boldsymbol{u}_m = \hat{u}_m + \Delta\boldsymbol{u}_m$$

Actual position difference between two sides computed on master side:

$$\Delta\boldsymbol{x}_{actual} = \boldsymbol{x}_m(t) - \boldsymbol{x}_{sd}(t - T)$$

Transmission may produce deflection between sides. Predict value of deflection assuming perfect initial conditions and no wave corrections:

$$\Delta\boldsymbol{x}_{predicted}(t) = \frac{1}{\sqrt{2b}}\int_0^t \boldsymbol{u}_m(\tau)\,d\tau$$

Position difference should reach zero in steady state when no further forces applied; i.e. when wave commands reach zero. Drift error:

$$\boldsymbol{d}(t) = \Delta\boldsymbol{x}_{predicted}(t) - \Delta\boldsymbol{x}_{actual}(t)$$

Derivative of drift error relates to corrective term $\Delta\boldsymbol{u}_m$:

$$\dot{\boldsymbol{d}}(t) = \frac{1}{\sqrt{2b}}\Delta\boldsymbol{u}_m$$

Adjust the wave command to change and reduce any existing drift errors:

$$\Delta\boldsymbol{u}_m = -\sqrt{2b}\,\boldsymbol{d}$$

Added path does not contain any power source or power flow. Corrective term may not introduce any power, for passivity criterion. Corrected wave command must be bounded by uncorrected term:

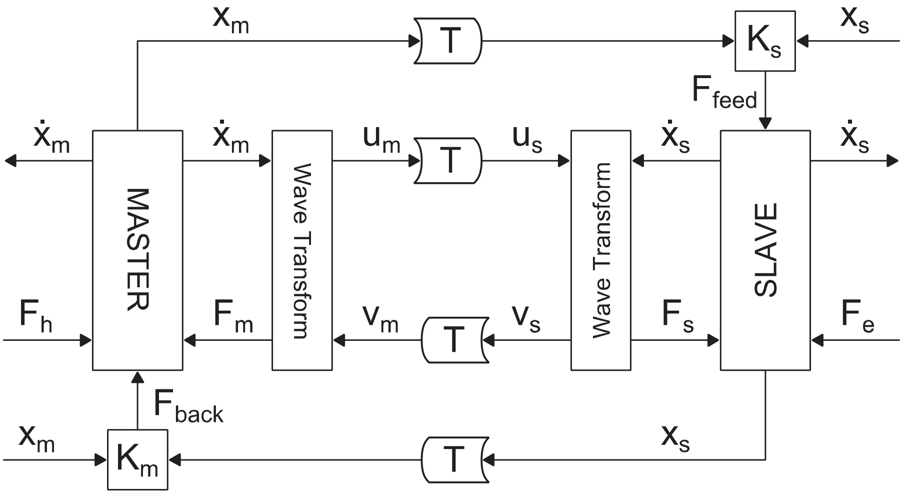
$$\frac{1}{2}\boldsymbol{u}_m^T\boldsymbol{u}_m \leq \frac{1}{2}\hat{u}_m^T\hat{u}_m$$

In order not to confuse operator, require both corrected and uncorrected wave command to be of same sign (for each DOF). Modulate magnitude of desired correction:

$$\Delta \boldsymbol{u}_m = \begin{cases} 0 & if \quad d\hat{\mathrm{u}}_m < 0 \\ -\sqrt{2b}\,\lambda \boldsymbol{d} & if \quad d\hat{\mathrm{u}}_m > 0 \quad and \quad \sqrt{2b}\,\lambda|\boldsymbol{d}| < |\hat{\mathrm{u}}_m| \\ -\hat{\mathrm{u}}_m & if \quad d\hat{\mathrm{u}}_m > 0 \quad and \quad \sqrt{2b}\,\lambda|\boldsymbol{d}| > |\hat{\mathrm{u}}_m| \end{cases}$$

and thereby assured of passivity and hence stability. Corrective term $\Delta\boldsymbol{u}_m$ and drift error $\boldsymbol{d}$ are never of same sign so that drift error will never increase. Error will decrease once system detects appropriate wave commands and associated power. System cannot force tracking until it receives some power input from operator. Bandwidth λ may be quite small as drift errors should be minimal.

<u>Bilateral Position exchange (BP)</u>; Scheme yields better position tracking and comparable force tracking with regard to traditional scattering scheme.



Master and slave modeled as mass-damper systems:

$$M_m\ddot{x}_m + B_m\dot{x}_m = F_h + F_{back} - F_m$$

$$M_s\ddot{x}_s + B_{s1}\dot{x}_s = F_s + F_{feed} - F_e$$

where M_m and M_s master and slave inertias, B_m and B_{s1} master and slave damping. F_h is the operator torque, F_e is the environment torque, and the other torques are defined as:

$$F_s = B_{s2}(\dot{x}_{sd} - \dot{x}_s)$$

$$F_{back} = K(x_s(t-T) - x_m)$$

$$F_{feed} = K(x_m(t-T) - x_s)$$

Position tracking error defined as:

$$e = x_m(t-T) - x_s(t)$$

In steady state:

$$F_{back} + F_{feed} = K(x_s(t-T) - x_m) + K(x_m(t-T) - x_s)$$

$$= -K\int_{t-T}^{t} \dot{x}_m(\tau) + \dot{x}_s(\tau)\, d\tau = 0$$

Or $F_{back} = -F_e$ which guarantees good force tracking on the master side.

G Simulation Results youBot arm

Position plots for master and slave objects when controlling the youBot arm.

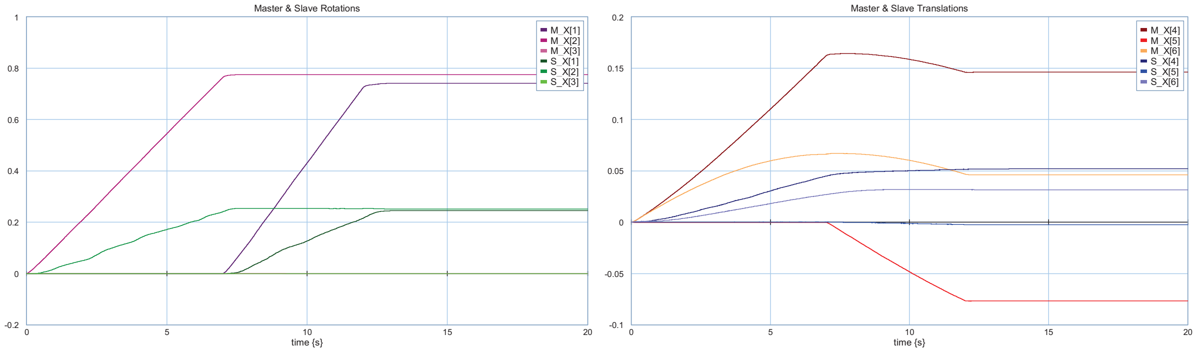


Figure G.6: Rotations (R_x, R_y, R_z) and translations (x, y, z) of Master & Slave objects without position control.

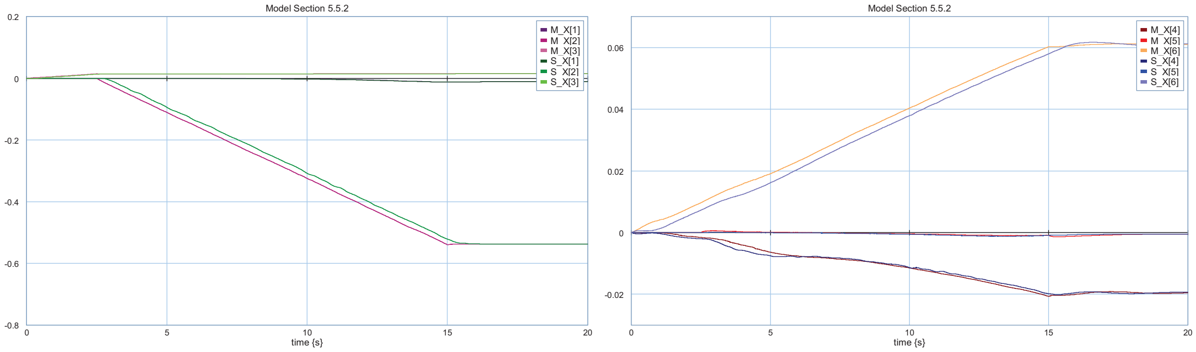


Figure G.7: Rotations (R_x, R_y, R_z) and translations (x, y, z) of Master & Slave objects with position control.

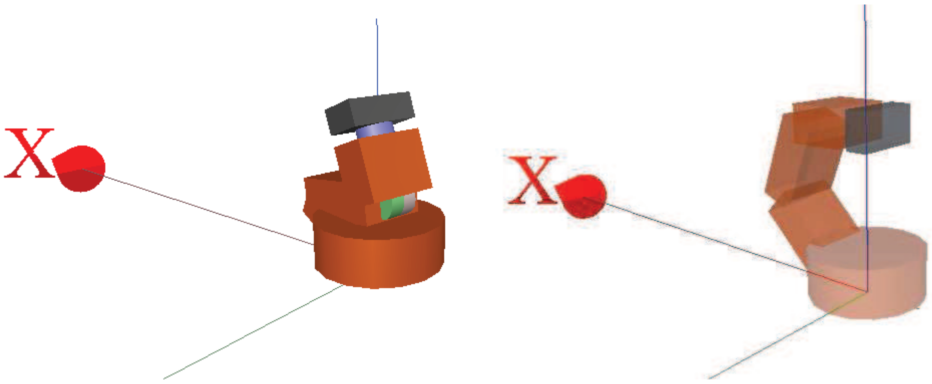


Figure G.8: a. Initial position of arm b. operating position of arm.

Bibliography

Alise, M.T. (2007), Expansion and Implementation of the Wave Variable Method in Multiple Degree-of-Freedom Systems, *Florida State University, Electronic Theses, Treatises and Dissertations, Paper 167*.

Alise, M.T., Roberts, R.G., Repperger, D.W., Moore, C.A. and Tosunoglu, S. (2009), On Extending the Wave Variable Method to Multiple-DOF Teleoperation Systems, *IEEE/ASME Transactions on Mechatronics, vol. 14, no. 1*, pp. 55-63.

Anderson, R.J. (1995), SMART: A Modular Control Architecture for Telerobotics, *IEEE Robotics & Automation Magazine, vol. 2, no. 3*, pp. 10-18.

Anderson, R.J. and Spong, M.W. (1989a), Asymptotic Stability for Force Reflecting Teleoperators with Time Delay, in *IEEE International Conference on Robotics and Automation, 1989, no. 3*.

Anderson, R.J. and Spong, M.W. (1989b), Bilateral Control of Teleoperators with Time Delays, *IEEE Transactions on Automatic Control, vol. 34, no. 5*, pp. 494-501.

Arcara, P. and Melchiorri, C. (2001), Control Schemes for Teleoperation with Time Delay; A Comparative Study, *Robotics and Autonomous Systems, vol. 38, no. 1*, pp. 49-64.

Aziminejad, A., Moallem, M. and Patel, R.V. (2006), A Study of Transparency in Wave-Based Teleoperation, in *IEEE International Conference on Mechatronics*.

Benedetti, C., Franchini, M. and Fiorini, P. (2001), Stable Tracking in Variable Time-Delay Teleoperation, in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 4*.

Breedveld, P.C. (1984), Physical Systems Theory in Terms of Bond Graphs, *PhD thesis, University of Twente, Enschede, Netherlands*.

Breedveld, P.C. (1985), Multibond–Graph Elements in Physical Systems Theory, *Journal of the Franklin Institute, vol. 319, no. 1/2*, pp. 1-36.

Brodskiy, Y. (2014), Robust Autonomy for Interactive Robots, *PhD Thesis, University of Twente, Enschede, Netherlands, ISBN 978-90-365-3620-2*.

Broenink, J. (1999), Introduction to Physical Systems Modelling with Bond Graphs.

Chiaverini, S., Siciliano, B. and Villani, L. (1999), A Survey of Robot Interaction Control Schemes with Experimental Comparison, *IEEE/ASME Transactions on Mechatronics, vol. 4, no. 3*, pp. 273-85.

Ching, H. and Book, W.J. (2005), Internet-Based Bilateral Teleoperation Based on Wave Variable with Adaptive Predictor and Direct Drift Control, *Journal of Dynamic Systems, Measurement, and Control, vol. 128, no. 1*, pp. 86-93.

Chopra, N., Spong, M.W., Ortega, R. and Barbanov, N.E. (2004), On Position Tracking in Bilateral Teleoperation, in *Proceedings of the 2004 American Control Conference, vol. 6*.

Cui, J., Tosunoglu, S., Roberts, R., Moore, C. and Repperger, D. (2003), A Review of Teleoperation System Control, in *Proceedings of the Florida Conference on Recent Advances in Robotics*.

Differ, H.G. (2010), Design and Implementation of an Impedance Controller for Prosthetic Grasping, *Mastersthesis, University of Twente*.

Dresscher, D., Brodskiy, Y., Breedveld, P. and Broenink, J. (2012), Modeling of the YouBot in a Serial Link Structure Using Twists and Wrenches in a Bond Graph, in *2nd International Conference on Simulantion, Modeling, and Programming for Autonomous Robots, SIMPAR 2010 Workshops*.

Franken, M., Misra, S. and Stramigioli, S. (2011), Improved Transparency in Energy-Based Bilateral Telemanipulation, *Mechatronics, vol. 22, no. 1*, pp. 45–54.

Franken, M., Stramigioli, S., Misra, S., Secchi, C. and Macchelli, A. (2011), Bilateral Telemanipulation With Time Delays: A Two-Layer Approach Combining Passivity and Transparency, *IEEE Transactions on Robotics, vol. 27, no. 4*, pp. 741-56.

Franken, M., Stramigioli, S., Reilink, R., Secchi, C. and Macchelli, A. (2009), Bridging the Gap Between Passivity and Transparency, in *Proceedings of Robotics: Science and Systems*.

Hogan, N. (1984), Impedance Control: An Approach to Manipulation, in *American Control Conference*.

Hokayen, P.F. and Spong, M.W. (2005), Bilateral Teleoperation; An Historical Survey, *Automatica, vol. 42, no. 12*, pp. 2035–57.

Lawn, C.A. and Hannafrod, B. (1993), Performance Testing of Passive Communication and Control in Teleoperation with Time Delay, in *Proceedings of IEEE International Conference on Robotics and Automation, vol. 3*.

Li, Z., Ding, L., Gao, H., Duan, G. and Su, C.-Y. (2013), Trilateral Teleoperation of Adaptive Fuzzy Force/Motion Control for Nonlinear Teleoperators With Communication Random Delays, *IEEE Transactions on Fuzzy Systems vol. 21, no. 4*, pp. 610-24.

Locomotec (2010), *KUKA youBot store*. [Online] Available at: http://www.youbot-store.com/.

Locomotec (2013), KUKA youBot User Manual.

Lozano, R., Chopra, N. and Spong, M.W. (2002), Passivation of Force Reflecting Bilateral Teleoperators with Time Varying Delay, *Proceedings of the 8. Mechatronics Forum*, pp. 24-30.

Malysz, P. and Sirouspour, S. (2011), Trilateral Teleoperation Control of Kinematically Redundant Robotic Manipulators, *International Journal of Robotics Research, vol. 30, no. 13*, pp. 1643-64.

Munir, S. and Book, W.J. (2001a), Wave-Based Teleoperation with Prediction, in *Proceedings of the 2001 American Control Conference, vol. 6*.

Munir, S. and Book, W.J. (2001b), Internet Based Teleoperation using Wave Variables with Prediction, in *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, vol. 1*.

Niemeyer, G. and Slotine, J.E. (1991), Stable Adaptive Teleoperation, *IEEE Journal of Oceanic Engineering, vol. 16, no. 1*, pp. 152-62.

Niemeyer, G. and Slotine, J.E. (2004), Telemanipulation with Time Delays, *The International Journal of Robotics Research, vol. 23, no. 9*, pp. 873-90.

Paynter, H.M. (1961), Analysis and Design of Engineering Systems, The M.I.T. Press, Boston: ISBN 0-262-16004-8.

Rodríguez-Seda, E.J., Lee, D. and Spong, M.W. (2006), An Experimental Comparison Study for Bilateral Internet-Based Teleoperation, in *IEEE International Symposium on Intelligent Control, International Conference on Control Applications, Computer Aided Control System Design*.

Scholl, K., Albiez, J. and Gassmann, G. (2001), MCA - An Expandable Modular Controller Architecture, in *Proceedings of the 4th Linux Real Time Workshop, Milano*.

Secchi, C., Stramigioli, S. and Fantuzzi, C. (2003), Digital Passive Geometric Telemanipulation, in *Proceedings of ICRA '03, IEEE International Conference on Robotics and Automation, vol. 3*.

Secchi, C., Stramigioli, S. and Fantuzzi, C. (2008), Transparency in Port-Hamiltonian-Based Telemanipulation, *IEEE Transactions on Robotics, vol. 24, no. 4*, pp. 903-10.

Secchi, C., Stramigioli, S. and Melchiorri, C. (2001), Geometric Grasping and Telemanipulation, in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3*.

Stanczyk, B. and Buss, M. (2005), Experimental Comparison of Interaction Control Methods for a Redundant Telemanipulator, in *11th IEEE International Conference on Methods and Models in Automation and Robotics*.

Stramigioli, S. (1998), A Novel Impedance Grasping Strategy Base on the Virtual Object Concept, in *6th Mediterranean conference; World Scientific, Control and systems*.

Stramigioli, S. (1998), From Differentiable Manifolds to Interactive Robot Control, *PhD Thesis, TU Delft, Delft, Netherlands, ISBN: 90-9011974-4*.

Stramigioli, S., Andreotti, S. and Melchiorri, C. (2000), Geometric Scattering in Tele-manipulation of Port Controlled Hamiltonian Systems, in *Proceedings of the 39th IEEE Conference on Decision and Control, vol. 5*.

Stramigioli, S. and Bruyninckx, H. (2001), Geometry and Screw Theory for Robotics, *eBook, Universita di Verona*.

Stramigioli, S., Melchiorri, C. and Andreotti, S. (1999), Intrinsically Passive Grasping and Manipulation, *IEEE Transactions of Robotics and Automation*.

Stramigioli, S., van der Schaft, A., Maschke, B. and Melchiorri, C. (2002), Geometric Scattering in Robotic Telemanipulation, *IEEE Transactions on Robotics and Automation, vol. 18, no. 4*, pp. 588-96.

Tanner, N.A. and Niemeyer, G. (2004), Practical Limitations of Wave Variable Controllers in Teleoperation, in *IEEE Conference on Robotics, Automation and Mechatronics, vol. 1*.

Wimböck, T., Ott, C. and Hirzinger, G. (2008), Analysis and Experimental Evaluation of the Intrinsically Passive Controller (IPC) for Multifingered Hands, in *IEEE International Conference on Robotics and Automation (ICRA)*.