Collision avoidance with haptic feedback for telepresence

An intelligent data management approach

Raziel Ivan Gonzalez Ramirez

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Collision avoidance with haptic feedback for telepresence

An intelligent data management approach

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Raziel Ivan Gonzalez Ramirez

November 12, 2015

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Collision avoidance with haptic feedback for telepresence

by

Raziel Ivan Gonzalez Ramirez

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: November 12, 2015

Supervisor(s): ____________________

dr.ir. M. Mazo Jr.

Reader(s): ____________________

dr. G.A. Delgado Lopes

____

dr. M.A. Zuñiga Zamalloa

____

Dr.-Ing. J. Kober

Abstract

The use of telepresence becomes an alternative in scenarios such as deep sea maneuvers, inspection and maintenance in remote and potentially hazardous environments, and aerial surveillance tasks for example. In some of these cases, human deployment might be dangerous, expensive and even not possible yet.

Because of the wide variety and increasing tasks in which remote robots are being used, there is a desire to reduce the learning curve for navigation with robots and create transparent systems. To achieve this, intuitive information such as haptic data and force feedback is proposed. Because of the characteristics of the remote environments and to prevent damage to the robot and the environment itself, collision avoidance algorithms able to perform safely under dynamic environments are required.

An additional challenge for the remote deployment of robots are the limitations in the communication channel. Information such as video, audio and control directives share the same communication channel and hence, the same bandwidth. It is important to have an intelligent data management system for the haptic information to prevent the saturation of such communication channels.

Table of Contents

Preface

The idea for this MSc thesis project came after performing an internship at Shell. During this internship, I became aware of the necessities of the Oil and Gas industry regarding robots: the control of such robots must be an intuitive task as the goal is to decrease the adoption time of remote robot technologies and this is directly linked to the complexity of such systems.

Haptics have been successfully introduced for dexterous and remote tasks in the past with robots such as the daVinci surgical robot. The steeper learning curve and adoption of these technologies are were possible because of the way they transmit the data, i.e. by making use of our natural way of communication: touch. After learning this, it became appealing to research and compare systems and technologies that will provide the haptic benefits while being able to cope with today's communication limitations, such as limited throughput velocities, delays, and latencies. Therefore, the aim of my research project is to assemble a set of algorithms to provide a safe remotely-operated robot navigation system. Moreover, one of the requirements of such a system is to reduce the workload on the operator.

Throughout the research, I came across several issues, therefore I would like to thank my supervisor dr.ir. M. Mazo Jr. for his support and assistance during the writing of this thesis.

Delft, University of Technology
November 12, 2015

Raziel Ivan Gonzalez Ramirez

Quiero dedicar este proyecto a mi familia, por ser la base de mi fortaleza y por siempre contar con su apoyo, sin importar cual fuera la decisión, con entera confianza de que ésta será la correcta.

A mi madre, Leticia Ramirez Azuara, por ser una mujer tan fuerte a quien admiro enormemente. Por su enorme apoyo y dedicación a lo largo de toda mi vida. Por siempre confiar en mí.

A mi padre, Joel Guillermo Gonzalez Bernal, por ser una persona para la cual cualquier problema siempre tiene solución. Me ha enseñado que siempre es posible encontrar lo positivo en todo. Por impulsarme.

A mi hermana, Maria Fernanda Gonzalez Ramirez, por empujarme a ser alguien mejor, aún cuando era atemorizante, siempre con una sonrisa y confianza en que el resultado será mucho mejor de lo esperado. Por su libre espíritu, el cual me ayudó a tomar la decisión de arriesgar, por explorar un mundo desconocido para mí.

To Anna Thysia Meijer, for her support and for helping me realize that things were not as hard as they may seem, bringing always a smile to my face. For believing in me.

Chapter 1

Introduction

The use of robotics has been exponentially increasing in the past thirty years because of the advantages they provide: the ability to perform repetitive tasks efficiently, they can be deployed in areas where humans are not able (because of safety and/or economical reasons), high availability as they can go through long working timeframes, and tailored designs for specific tasks. Moreover, robots are able to provide us with access to remote and/or hazardous areas through telepresence, enabling the interaction with previously unreachable environments.

Several industries can benefit of the use of robots, for example the Oil and Gas (OG) industry. New challenges will be faced with fields located at harsh and remote areas, with are challenging for human deployment due to elevated human costs deriving from sophisticated security equipment, wages, and upkeep; Health, Safety, Security and Environment (HSSE) frameworks are implemented to prevent risk situations at the work environments, making the work environment safer which also increases the requirements and may increase the costs.

Because of the aforementioned reasons, high up-time and lower upkeep costs are key factors for a successful OG industry as this could represent several advantages to the business. To achieve this, robots and automation have been used both to improve HSSE conditions and performance in a number of industries over the last three decades; robots are able to perform repetitive tasks quickly and efficiently, go through longer working hours, deal with hazardous materials and keep people out of potentially dangerous environments. Advantages such as the reduction of human exposure, upkeep, traveling costs, and making information instantly available to experts all over the world while being able to interact with the remote environment, are directly derived from the use and implementation of robots.

In order to preserve safety, for the robot and the surrounding environment, a collision avoidance system implemented in the remote robots is a necessary function to prevent collisions and damage. Because of the characteristics of the environment in which robots are deployed, e.g. unknown, dynamic and complex environments, a robust collision avoidance system is designed.

In this project, a shared control scheme is proposed: an *operator* is in control of the robot and its navigation throughout the environment, based in visual, auditory and haptic feedback.

The haptic feedback data is generated making use of the collision avoidance algorithms, and in case of an imminent risk situation, the system must take over control and take action to preserve safety.

1-1 Motivation

In the previous section, the advantages in the implementation of robots in remote environments have been discussed. Despite of these advantages being enhanced with a fully-automatic control scheme, human operators are still required in order to make decisions based on the information obtained by the sensing devices mixed with the expertise and experience. Due to this desired operator profile, the control must be an intuitive and easy task; to provide an intuitive method for controlling the robot, haptic data is greatly desired as this is the innate method of communication for human beings, and furthermore training times for remote control can be reduced through the implementation of such information.

The use of telepresence becomes an alternative to human deployment in scenarios such as deep sea maneuvers, inspection and maintenance in hazardous environments or surveillance tasks. The OG industry, has an specific interest in this technology as the scope is to achieve unmanned facilities, with the advantages previously described and furthermore, the possibility to work on previously unreachable areas. The use of robotics in the OG industry is growing as the oil fields are being located in remote and unwelcoming environments.

The limitations delay impose over a system as the one described before, are that it is impossible for a command to be applied on the system in less than a one-way delay time T. Furthermore, the reaction would take at least $2T$, limiting the system's bandwidth. Due to this, delay represents a major impact not only on the system's performance but also on the workload and focus of the operator. Because of this, it is greatly desired that the system's performance is able to cope with delays without representing an extra task to the operator.

1-2 Goal of the thesis

The goal of this thesis is to compare collision avoidance algorithms for a shared-control scheme for telepresence. Such a system provides haptic feedback in the form of attractive and repulsive forces on a haptic interface device, e.g. a steering wheel. Moreover, the inclusion of the haptic data must not represent an extra payload on the communication channel, thus haptic data compression is required; the data compression method must preserve the system's transparency, even under delayed communications. These requirements aim for a simple and intuitive system.

Additionally, this project also seeks to test a collision avoidance algorithm, based on Vector Field Histograms, as a force feedback generator. A successful implementation could represent computational savings as no dedicated force generator algorithm would be required.

1-3 Current state of Haptic technologies

Research on remote robotics for different industries is ongoing with the same goal: *transparency*. Industries such as the Aerospace, Oil and Gas, and Medical Industry, have shared challenges: hazardous, remote or unwelcoming environments, long distance communication, limited resources and elevated transport costs. Because the use of robotics and haptics has been used as an alternative to the aforementioned problems, now the aim of the research is to increase the transparency, as it linearly improves the performance of the operators. For example, the European Space Agency (ESA) has ongoing projects dedicated to haptic robots for dexterous activities with the Telerobotics Lab such as 'Haptics 1' [1].

Telepresence to solve complex (dexterous) tasks is another alternative for the use of such robots. The DaVinci surgical robot, developed by Intuitive Surgical Inc. [2], is amongst the first widely distributed telepresence systems able to solve dexterous tasks. Several studies have been performed on the challenges derived from the usage of such a telepresence system, e.g. training times and technology adoption. All of them conclude that the inclusion of haptics, decreases substantially the training times, which therefore decrease the technology adoption time [3], [4], [5].

1-4 Thesis organization

Chapter 2 covers the preliminaries required for this project.

Chapter 3 elaborates on the topics: *Collision avoidance and Force Feedback* and *Intelligent Data Managment*, presenting the solutions to be compared in this project. The studied methods for a collision avoidance algorithm are extensively described explaining the relevancy with the project. The methods here described form the theoretical basis of the project and its evaluation. Additionally, the methods to generate repulsive forces based on the obstacles in the environment are described in this chapter as well. Finally, this chapter covers the methods that allow a reduction on the communication footprint caused by the haptic data addition under the `intelligent data management` scheme. These methods show how the system is able to perform under compressed and delayed communications.

Chapter 4 describes the experimental setup: the robot, maps and control parameters used to build a setup for comparison.

Chapter 5 presents the results of the comparison tests. A discussion is presented with the goal to provide the reader with an insight on the results here achieved.

Chapter 6 Based on the findings on Chapter 5, a set of conclusions and recommendations are derived.

Chapter 2

Preliminaries

The aim of this chapter is to bring these problems into a mathematical context, basis for the solutions compared in this project. First, a context is created by introducing the topic of teleoperation.

Next, current ongoing research in the area of virtual forces is presented, followed by the two *collision avoidance and force feedback* methods selected for this project, Parametric Risk Field (PRF) and Vector Field Histogram + (VFH+). These methods, will convert the characteristics of the remote environment captured by the sensors into kinesthetic information, i.e. touch information.

Next in this chapter, covers the basis of *event-based control* as a brief introduction to the *intelligent data management* approach of this project in order to achieve *haptic data compression.* This is followed by dissipativity and passivity theory in order to cope with the delays on to propose a solution to communication delays based in an energy-supervision scheme.

The modeling for a differential-drive robot is presented, as they are the basis for the simulations for this project, and finally, a sketch of the problem to solve is presented.

2-1 Telepresence

Telepresence systems have been growing exponentially in the past few years due to the great advantages that they represent for humans. Telepresence systems allow humans to be present and active in environments which were not possible before; remote, hazardous or simply on a different scale are frequent characteristics for such environments. Because of this, Telepresence and teleaction systems have been subject of extensive and ongoing research. Figure 2-1 shows an example of a basic Telepresence system with its three main characteristics:

1. the Human-System Interface (HSI),

2. the teleoperator,

3. and the communication channel

The HSI performs both as the data input, e.g. keyboards, joysticks and haptic devices for position/orientation, and as the data output, e.g. video feeds, auditive and haptic feedback. The robot is equipped with multiple sensors and actuators which enables interaction with the environment. Finally, the communication link transports the multimodal data streams bidirectionally.

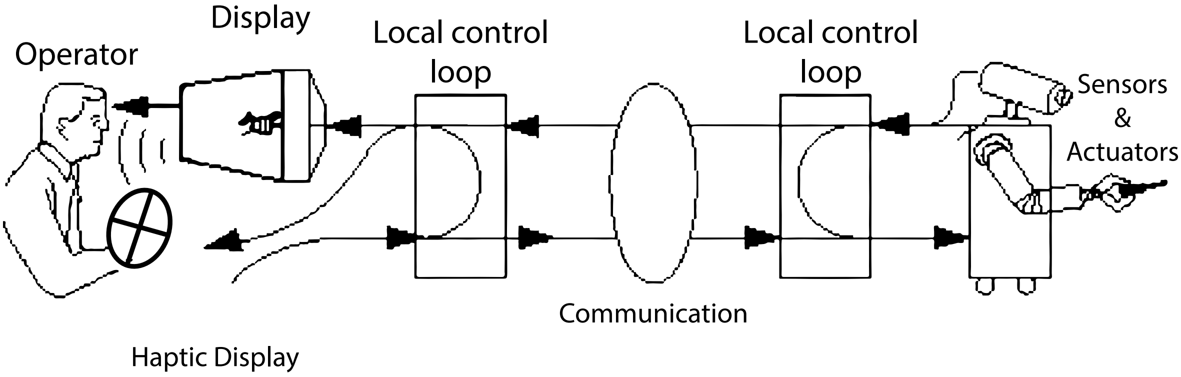


Figure 2-1: General overview of a telepresence system [6]

Ensuring that the same kind of information becomes available regardless of the location of the operator is one challenge here introduced. This is one of the goals of the current project and it is defined with the name of *transparency*, defined by the authors of [7] as the design goal of teleoperation, i.e. creating an immersive environment such that "the human operator cannot distinguish between direct interaction with the environment and tele-operated interaction". Such systems are called *transparent*.

Because the intended use of such robots will be deployed in complex environments, e.g. environments with pipelines, rocks, stairs, and unknown objects, etc., high safety precautions must be considered in order to prevent risk situations that may be derived from the use of remote robotics therefore, a collision avoidance system is desired.

An important thing to consider for these scenarios is the restricted communication channel. This is due to the fact that the facilities for which these robots are intended are usually remotely located and therefore communication is a highly valuable resource. Because of this, the addition of haptic data should not represent an elevated cost for communications; optimized and minimal haptic data information is desired while preserving the transparency of the control interface.

2-2 Differential-drive robot

Differential drive robots, as the one shown in Figure 2-2 are non-linear systems modeled by Eq. (2-1) to (2-7) that reside in the Special Euclidean group $SE(2)$. In these equations, ω_r and ω_l are the angular velocities of the wheels. As it can be seen, the mapping between angular velocities of the wheels and the linear velocity of the robot cab be easily performed.

However, it is of common practice to use a simpler model for design purposes. Therefore the unicycle model is used. This is shown in Equations (2-1), (2-2), and (2-3).

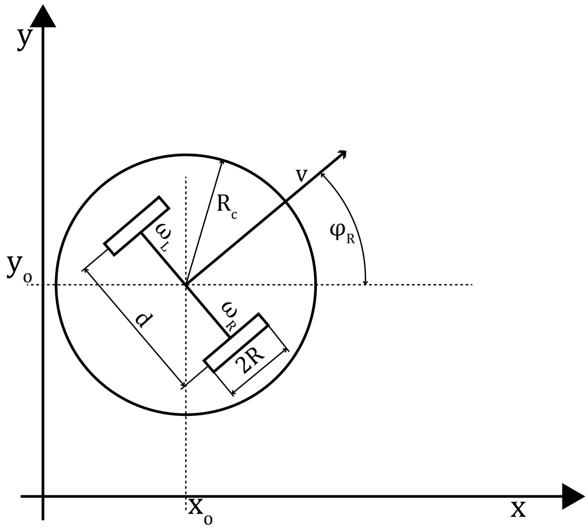


Figure 2-2: Differential drive modeling

$$\dot{x}_o = v \cdot \cos \varphi_R \tag{2-1}$$

$$\dot{y}_o = v \cdot \sin \varphi_R \tag{2-2}$$

$$\dot{\varphi}_R = \omega \tag{2-3}$$

Through the integration of Equations (2-1 − 2-3), we can obtain the robot's configuration (pose) $[x_o,\ y_o,\ \varphi_R]^T$, however the control characteristics cannot be implemented through this parameters, as the controllable parts of the robot are the angular velocities of the wheels, i.e. ω_l and ω_r. By defining v and ω as in Eq. (2-4) and Eq. (2-5) and substituting in Equations (2-1 − 2-3) we can obtain the models shown in Eq. (2-7) and Eq. (2-7), which can be implemented:

$$v = \frac{R}{2}\left(\omega_r + \omega_l\right) \;\Rightarrow\; \begin{cases} \dot{x} = \frac{R}{2}\left(\omega_r + \omega_l\right)\cos\varphi_R \\ \dot{y} = \frac{R}{2}\left(\omega_r + \omega_l\right)\sin\varphi_R \end{cases} \tag{2-4}$$

$$\omega = \frac{R}{d}\left(\omega_r - \omega_l\right) \;\Rightarrow\; \dot{\varphi}_R \tag{2-5}$$

$$\omega_r = \frac{2v + \omega d}{2R} \tag{2-6}$$

$$\omega_l = \frac{2v - \omega d}{2R} \tag{2-7}$$

The compact version of the non-linear system in Equations (2-1 to 2-7), is then written as:

$$\Sigma : \begin{cases} \dot{x} = f(x, u) \\ y = h(x, u) \end{cases} \tag{2-8}$$

where:

- $x = [x_o,\ y_o,\ \varphi_R]^T \in \mathbb{R}^3$ denotes the state of the robot,
- $u = [\omega_l,\ \omega_r]^T \in \mathbb{R}^2$ represents the control input to the robot

2-3 Virtual Forces

In order to properly avoid collisions through telepresence, several methods have been derived and studied for this specific task, all of them with different characteristics. Whether it is a Unmanned Aerial Vehicle (UAV), deep sea or a ground robot, a common approach has been studied: the use of virtual forces to represent the remote obstacles through haptic feedback. How these forces are generated greatly varies on the method chosen for it. In the study performed by [8], the haptic collision avoidance is intended for a wheelchair, providing haptic feedback to the user about the surroundings as a function of distance. In [9], the authors present a deeper and highly relevant discussion on different methods to generate these virtual forces. They take into account the force's magnitude in order to use it properly into force feedback devices as well. The first discussion is regarding the Artificial Force Field (AFF) and its known problems: difficult passage through narrow corridors and local minima in the distance with objects, this could cause a trap for the robot. Generalized Potential Field (GPF) suggested by [10], an artificial potential field based on the distance between the robot and the obstacle is generated. The repulsive force is then generated by taking the gradient of this potential field. In [8, 11], a virtual force is generated as a function of the distance between the robot and the obstacles in the surrounding environment.

A popular method in teleoperation to create virtual forces is through the use of AFFs, as first introduced by [12]. This method creates an imaginary area surrounding the robot, based on different characteristics such as the distance, the velocity vector, and the certainty of the objects on the remote environment. These methods have been used successfully in shared-control schemes. [13, 14].

Figure 2-3 shows an example on how the potential fields can be interpreted as hills, driving away the robot (repelling it), this can be a simple interpretation of the way it works. Goals, can be represented as sinkholes, representing a smaller cost (effort) compared to going uphill to the obstacles

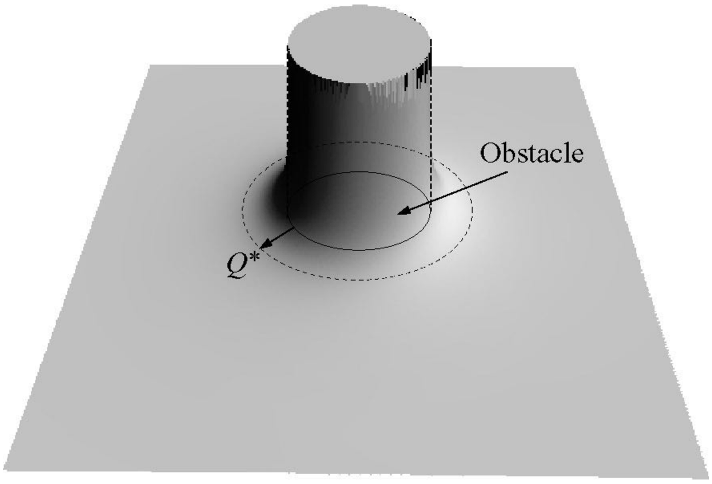


Figure 2-3: Virtual force created for a cylinder-shaped obstacle [15]

2-3-1 Artificial force field

Feedback force

The purpose of an AFF, also referred to as "potential field", is to translate the characteristics of an environment into haptic data through imaginary forces. In this framework, obstacles are represented as repulsive forces and in the presence of a goal or target, this could be represented as an attractive force.

To compute these forces, a risk field is created around the robot with customizable parameter, which allows a parametric adjustment of the risk field (area where no obstacles are allowed) as proposed by the authors of [16] with the name PRF. This method uses the distance from the obstacle to the robot and the current velocity of the robot to generate the repulsive force.

Generating the risk field

The risk field provides an imaginary protection area in which the obstacles are not allowed and it is a function of the velocity of the robot. The parametrized risk field is shown in Figure 2-4. The critical region, i.e. where no object is allowed (shown in gray), consists of the protection radius r_{pz} with a region along the velocity vector $\mathbf{v}$. The length of this region is defined as the stopping distance of the vehicle, obtained from the velocity $\mathbf{v}$ and the maximum deceleration of the robot a_{max} as shown in Eq. (2-9).

$$d_{stop} = \frac{|\mathbf{v}|^2}{2\ a_{max}} \tag{2-9}$$

The outline if the PRF is determined as a function of a minimum distance from the critical region d_{min}. The field is then extended by a distance d_{ahead} which depends on $\mathbf{v}$ and the time constant t_{ahead} shown in Eq. (2-10), which allows the operator to react to obstacles; this time constant is empirically determined.

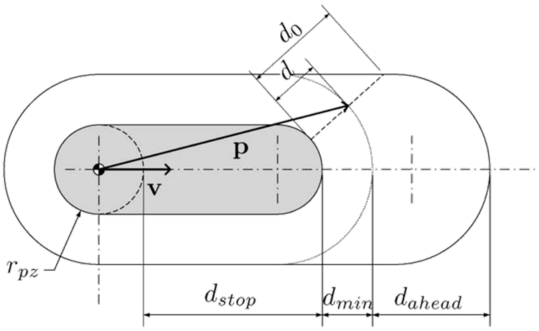


Figure 2-4: PRF: parameter and variable definitions.

$$d_{ahead} = |\mathbf{v}|\ t_{ahead} \tag{2-10}$$

For all the obstacles that are located outside the outline the risk is zero. To compute the risk value for point $\mathbf{p}$, two variables are computed: d and d_0, d is described as the minimum distance from the point $\mathbf{p}$ to critical area outline and d_0 is the distance between the boundary of the critical region and the outline. With this, the risk value $P(\mathbf{p}, \mathbf{v})$ for point $\mathbf{p}$ is determined as shown in Eq. (2-11):

$$P(\mathbf{p}, \mathbf{v}) \begin{cases} 0 & \text{if outside the outline} \\ 1 & \text{if inside critical region} \\ p(d/d_0) & \text{otherwise} \end{cases} \tag{2-11}$$

To determine how the risk varies inside the PRF, the function $p(d/d_0)$ can be freely selected. Some examples given in literature involve a linear function (Eq. (2-12)), a cosine (Eq. (2-13)) or a shifted cosine function (Eq. (2-14)).

$$p(d/d_0) = 1 - \frac{d}{d_0} \tag{2-12}$$

$$p(d/d_0) = \frac{1}{2}\cos\left(\frac{d}{d_0}\pi\right) + \frac{1}{2} \tag{2-13}$$

$$p(d/d_0) = \cos\left(\frac{d}{d_0}\frac{\pi}{2} + \frac{\pi}{2}\right) + 1 \tag{2-14}$$

Determining the risk direction

The theory determines two possibilities to determine the risk direction:

1. **radial:** The risk vector points in the direction of the center of the vehicle, shown in Figure 2-6 (a).

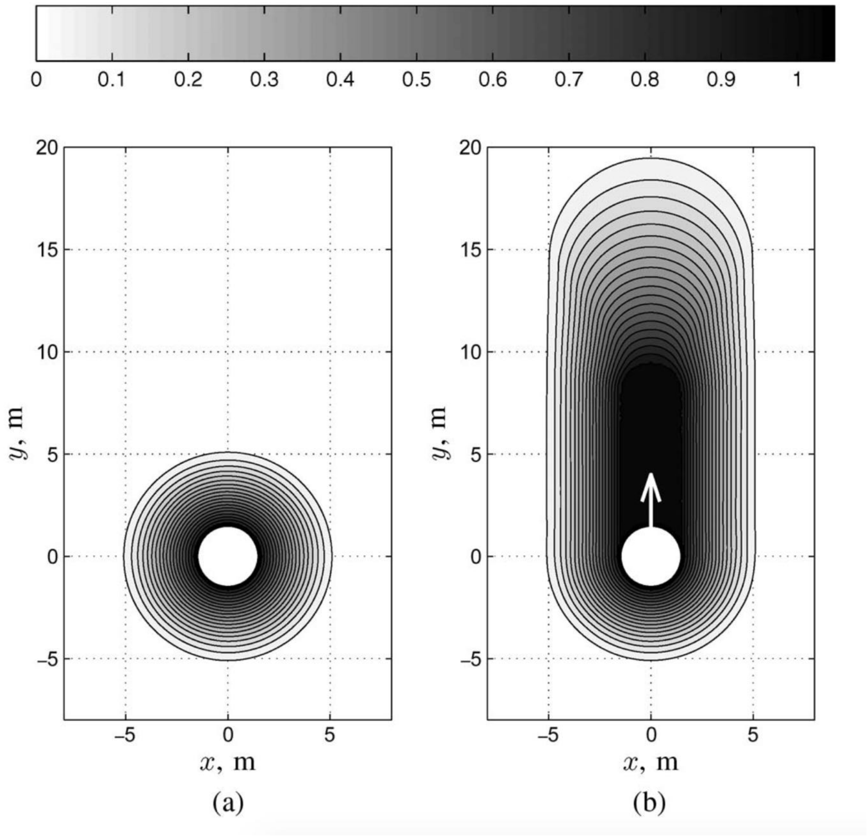


Figure 2-5: PRF contours for two different velocities: *a)* $\mathbf{v} = 0$ and *b)* $\mathbf{v} = 4$ m/s with $r_{pz} = 1.5$ m, $a_{max} = 1$ m/s^2, $d_{min} = 4.5$ m $(= 3\ r_{pz})$, and $t_{ahead} = 2\ s)$.

2. **normal:** The risk vector is mapped perpendicular to the object's surface, shown in Figure 2-6 (b).

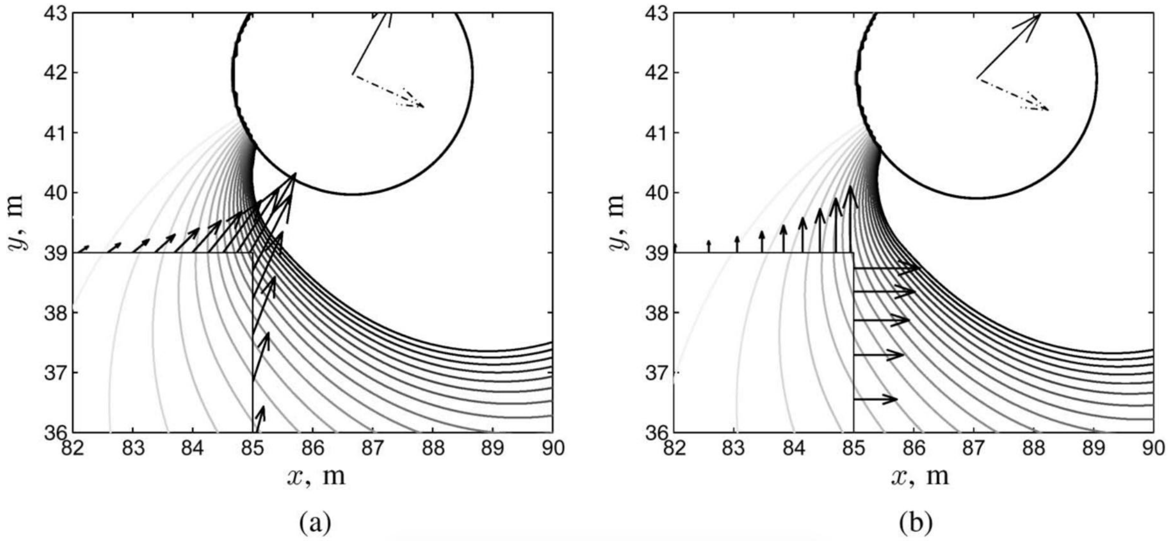


Figure 2-6: PRF: risk direction.

Based on the discoveries achieved in [9], the best haptic data for narrow passages was obtained through the implementation of the radial direction. Even though the navigation is not focused on the navigation in narrow passages, the plausibility of this is reason enough to choose for this method.

Summing multiple obstacles

The risk vectors calculated from each sensor need to be integrated into one final risk vector before transforming it into an artificial repulsive force vector. To achieve this, there are three ways:

1. *sum:* the individual risk vectors are added, creating a final risk vector. With this method, low risks could be overestimated.
2. *mean:* the mean of all the risk vectors is taken. With this method a big risk could be underestimated.
3. *max, min:* the sum of the largest positive and negative risk vectors with the final avoidance vector limited to one. This method was chosen to be used in this study.

Finally, the repulsive force F_s can be computed as:

$$F_s = P(\mathbf{p}, \mathbf{v}) \cdot \psi$$

where ψ is a user-defined scaling factor.

2-3-2 Vector Field Histogram +

In their research, the authors of [17] improved their previous research on Virtual Force Fields, namely the Vector Field Histogram. This new and improved version named Vector Field Histogram +, offers improvements that result in smoother robot trajectories and increased reliability. The method is explained below:

1. Cartesian grid A two-dimensional cartesian grid is created. The value $c(i,j)$ contains information about the certainty of the existence of an object in the cell (i,j). As the robot moves, a circular window of radius w_s moves with it, called the active region $\mathcal{C}^*$; active cells corresponding to the active region are denoted by $c^*_{i,j}$.

The process of the VFH+ compels the following data reduction steps:

2. Polar histogram The first data reduction step maps the active region $\mathcal{C}^*$ of the grid $\mathcal{C}$ into the polar histogram $\mathcal{H}^p$. The content of each active cell $c^*_{i,j}$ is treated as an obstacle vector. The vector direction $\beta_{i,j}$ is determined by the direction of the active cell to the Robot Center Point (RCP) as shown in Eq. (2-15).

$$\beta_{i,j} = \arctan\left(\frac{y_j - y_o}{x_i - x_o}\right) \tag{2-15}$$

where: $(x_o, y_o) \equiv r_{x,y}$, are the coordinates of the RCP, and (x_i, y_j) are the coordinates of the active cell $c^*_{i,j}$. Once the angle has been computed, the vector magnitude $m_{i,j}$ of an active cell $c^*_{i,j}$ is given by Eq. (2-16) (shown below again):

$$m_{i,j} = c^2_{i,j}\left(a - bd^2_{i,j}\right) \tag{2-16}$$

where $c_{i,j}$ and $d_{i,j}$ are the certainty value and distance to the RCP of cell $c^*_{i,j}$ respectively, and the parameters a and b are chosen according to Eq. (2-17):

$$a - b\left(\frac{w_s - 1}{2}\right)^2 = 1 \tag{2-17}$$

By squaring $c^2_{i,j}$, the confidence that high certainty values represent obstacles is shown, as opposed to low certainty values which may be caused by noise. The vector magnitude also takes into account the distance from the obstacle to the robot with $d_{i,j}$, closer obstacles represent higher risks and therefore higher magnitudes. Up until now, this method differs in nothing from the Vector Field Histogram (VFH).

The polar histogram $\mathcal{H}^p$ has an arbitrary angular resolution α so that $n = 360°/\alpha$ is an integer. Therefore, there are n number of sectors k corresponding to a discrete angle $\rho = k\ \alpha$. To compensate for the width of the robot, the VFH+ method uses a theoretically determined low-pass filter, i.e. enlarging the obstacle cells by the robot radius (r_r), defined as the maximum distance from the RCP to its perimeter; non-circular robots can be modeled as a circle with the radius as the longest distance from its center depending on the orientation of the vehicle [18]. For safety, a minimum distance d_s between the robot and the obstacle is added, therefore the obstacle cells are enlarged by $r_{r+s} = r_r + d_s$. With this, the robot can be treated as a point-like vehicle.

For each cell, the enlargement angle $\gamma_{i,j}$ is defined by Eq. (2-18).

$$\gamma_{i,j} = arcsin\left(\frac{r_{r+s}}{d_{i,j}}\right) \tag{2-18}$$

Therefore, the polar obstacle density $\mathcal{H}^p$ can be defined for each sector as shown in Eq. (2-19).

$$\mathcal{H}^p_k = \sum_{(i,j)\in\ \mathcal{C}} m_{i,j} \cdot h^{'}_{i,j} \tag{2-19}$$

where:

$$h^{'}_{i,j} = \begin{cases} 1 & \text{if } k \cdot \alpha \in [\beta_{i,j} - \gamma_{i,j},\ \beta_{i,j} + \gamma_{i,j}] \\ 0 & \text{otherwise} \end{cases} \tag{2-20}$$

Additionally, the findings from the previous section can be used here as well to compute the polar obstacle density such that Eq. (2-21) can be used replacing Eq. (2-19), as proposed by [19]. The advantage of this method is that the maximum risk will not be underestimated and a minimum risk will not be overestimated.

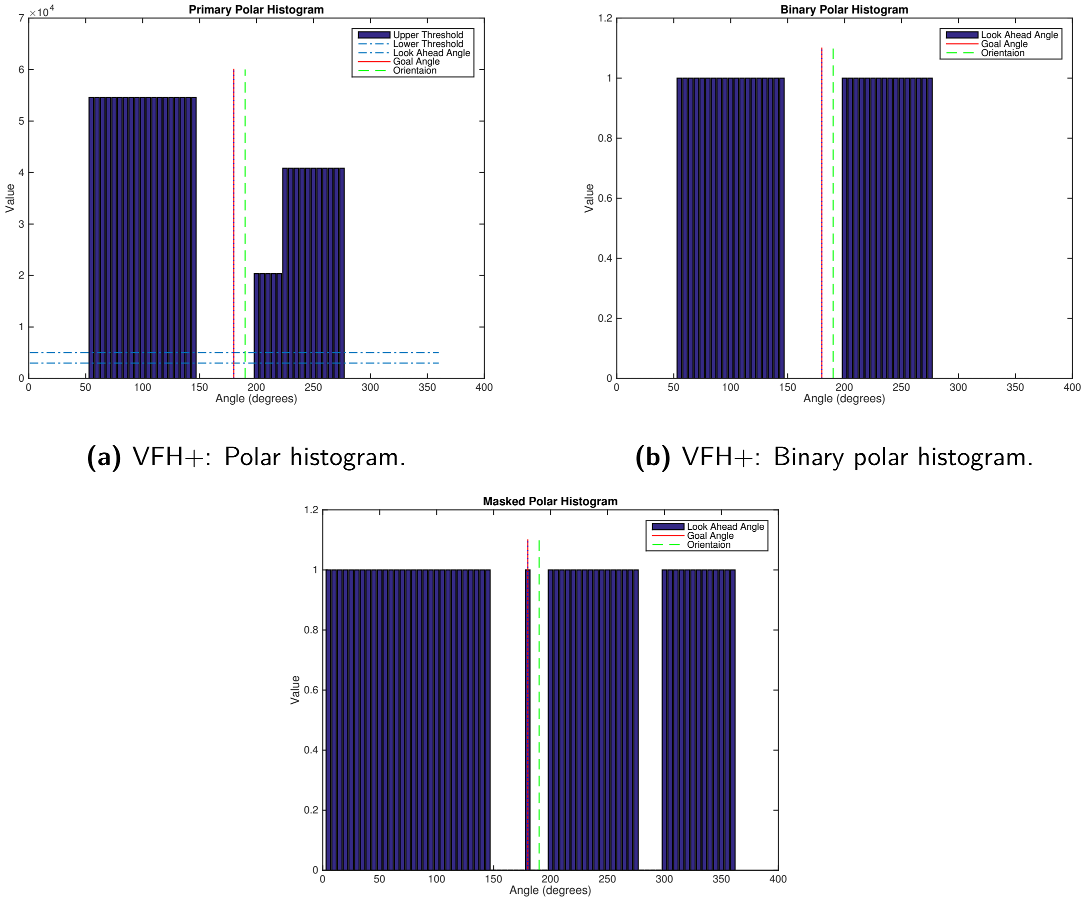
$$\mathcal{H}^p_k = \max\left(m_{i,j}\right) \cdot h^{'}_{i,j} \tag{2-21}$$

An example of the polar histogram is shown in Figure 2-7a.

3. Binary polar histogram The second stage of data reduction brings a smoother trajectory due to the avoidance of oscillations in the steering commands. This is particularly noticeable in environments with narrow openings, e.g. oscillations are present due to closeness to both sides of the opening. This resembles a nervous system which is "not committed to the decision". To reduce this behavior, a hysteresis approach is follows by creating two thresholds, τ_{low} and τ_{high}. With the data provided by the first histogram, $\mathcal{H}^p$ and these thresholds, the binary polar histogram $\mathcal{H}^b$ is created. By replacing the polar density for a binary value, this polar histogram indicates which directions are free for a robot. To build it, the rules shown in Eq. (2-22) must be followed. This is shown in Figure 2-7b.

$$\mathcal{H}^b_{k,i} = \begin{cases} 1 & \text{if } \mathcal{H}^p_{k,i} > \tau_{high} \\ 0 & \text{if } \mathcal{H}^p_{k,i} < \tau_{low} \\ \mathcal{H}^p_{k,i-1} & \text{otherwise} \end{cases} \tag{2-22}$$

4. Masked polar histogram To account for the robot dynamics and kinematics, while constructing the third histogram, it is not assumed that the robot is able to switch its direction of travel instantly, as shown in Figure 2-8. It assumes that the robot's trajectory is based on circular arcs as a function of the robot velocity defined by $\kappa = 1/r$. This becomes highly relevant



(a) VFH+: Polar histogram.

(b) VFH+: Binary polar histogram.

(c) VFH+: Masked polar histogram.

Figure 2-7: VFH+ histograms

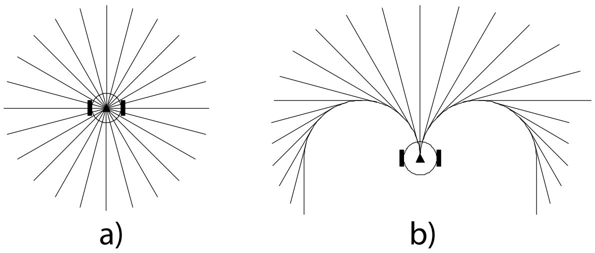
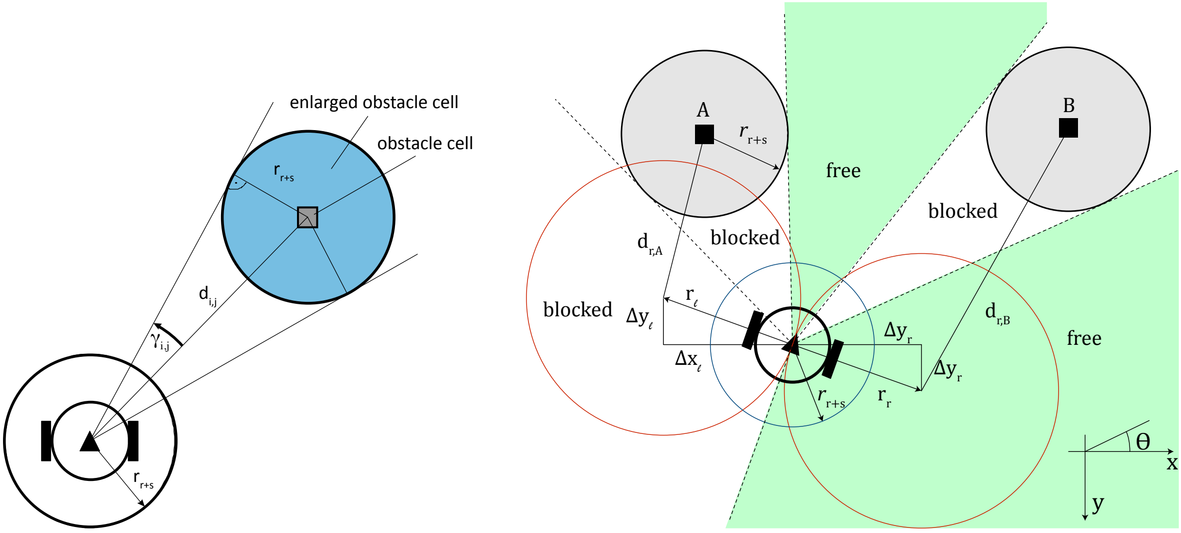


Figure 2-8: VFH+: Approximation with robot dynamics: a) differential drive, b) Ackerman steering.

for vehicles based on the *Ackerman steering* model or the *tricycle* mechanism. Furthermore, the radius for both sides can be modeled as $r_r = 1/\kappa_r$ and $r_l = 1/\kappa_l$.

Once these parameters are known and with the map grid, the blocked sectors, i.e. the ones that will collide with the circular trajectories, can be easily detected. To take into account the size of the robot, the obstacles are enlarged again by r_{r+s}. Figure 2-9b shows an example of blocked directions.



(a) VFH+: Enlargement angle.

(b) VFH+: Example of blocked directions.

Figure 2-9: VFH+

The positions of the trajectory centers relative to the current robot position are defined as shown in Eq. (2-23):

$$\begin{aligned} \Delta x_r &= r_r \cdot \sin(\theta) \quad & \Delta y_r &= r_r \cdot \cos(\theta) \\ \Delta x_l &= -r_l \cdot \sin(\theta) \quad & \Delta y_r &= -r_l \cdot \cos(\theta) \end{aligned} \tag{2-23}$$

The distances from the active cell $c^*_{i,j}$ to the two trajectory centers is given by Eq. (2-24):

$$\begin{aligned} d_r^2 &= (\Delta x_r - \Delta x(i))^2 + (\Delta y_r - \Delta y(j))^2 \\ d_l^2 &= (\Delta x_l - \Delta x(i))^2 + (\Delta y_l - \Delta y(j))^2 \end{aligned} \tag{2-24}$$

Now, with this data we can determine which side of the robot is blocked by obstacles that intersect the circular trajectories by Eq. (2-25):

$$\begin{cases} \text{if } d_r^2 < (r_r + r_{r+s}) & \text{an obstacle blocks the direction to its right} \\ \text{if } d_l^2 < (r_r + r_{r+s}) & \text{an obstacle blocks the direction to its left} \end{cases} \tag{2-25}$$

By checking this condition, two limit angles can be withdrawn of it, namely φ_r for right angles and φ_l for left angles. The base limit angle $\varphi_b = \theta + \pi$ is defined as the direction opposite to the current motion. In case all trajectories are blocked, this represents the course of action. The authors of [17] then describe the algorithm that should be followed to efficiently implement the method:

1. Determine φ_b. Set φ_r and φ_l equal to φ_b.

2. For every cell $c_{i,j}$ in the active window $\mathcal{C}$ with $c_{i,j} > \tau$:
   (a) If $\beta_{i,j}$ is to the right of θ and to the left of φ_r, check the first term in Eq. (2-25). If the condition is not satisfied, set φ_r equal to $\beta_{i,j}$.
   (b) If $\beta_{i,j}$ is to the left of θ and to the right of φ_l, check the second term in Eq. (2-25). If the condition is satisfied, set φ_l equal to $\beta_{i,j}$.

With the values $\mathcal{H}^b$, φ_r and φ_l, the masked polar histogram is built as follows in Eq. (2-26):

$$\mathcal{H}_k^m = \begin{cases} 0 & \text{if } \mathcal{H}_k^b = 0 \text{ and } k \cdot \alpha \in \{[\varphi_r, \theta], [\theta, \varphi_l]\} \\ 1 & \text{otherwise} \end{cases} \tag{2-26}$$

The masked polar histogram shows the available directions at the current speed. If all directions are blocked, the robot will have to slow down and determine a new values for φ_r and φ_l based on a lower speed. If still all values are blocked, the robot would have to stop completely. This can work as a trap detection algorithm. The output of each of this polar histograms is shown in Figure 2-7c.

5. Selection of the steering direction Due to the possibility that the masked polar histogram may have more than one feasible steering direction, a final assessment on these options must be performed. The original VFH proposed in [20] is goal oriented. The introduction of a cost function in [17] takes into account the candidate direction k_n with the lowest cost to be the new direction of motion $\varphi_n = \alpha k_n$.

First, the candidate openings are categorized between *wide* and *narrow* sections. An opening is considered wide if the difference between the borders is bigger than s_{max} sectors. Otherwise the opening is considered narrow. For a narrow opening, there is only one candidate direction, i.e. through the center of the gap:

$$c_n = \frac{k_r + k_l}{2} \tag{2-27}$$

For a wide opening, there are three candidate directions: leaning to the left or to the right of the opening, and the target direction if it lies in between the two of them. The candidate directions c_l and c_r drive the robot at a safe distance from the obstacle while c_t drives it towards the goal.

$$\begin{aligned} c_r &= k_r + \frac{s_max}{2} \\ c_l &= k_l - \frac{s_max}{2} \\ c_t &= k_t \qquad\qquad \text{if } k_t \in [c_r, c_l] \end{aligned} \tag{2-28}$$

The best suitable direction is determined by the cost function g in Eq. (2-29).

$$\min \left(g(c) = \mu_1 \cdot \Delta(c, k_t) + \mu_2 \cdot \Delta\left(c, \frac{\theta_i}{\alpha}\right) + \mu_3 \cdot \Delta(c, k_{n,i-1}) \right) \tag{2-29}$$

- The first term of Eq. (2-29) represents the cost associated with the difference between the candidate direction and the target direction. This is the goal oriented behavior.
- The second term is the cost associated to the difference between the candidate direction and the robot's wheels orientation. This resembles mechanical memory and commitment to a direction.
- The third term is the cost associated to the difference of the new candidate direction and the previous one $\varphi_{t,i-1}$. This can create a smoother robot trajectory because of the commitment to a direction.

The values for $\mu_1 = 5$, $\mu_2 = 2$ and $\mu_3 = 2$ are suggested by the authors for a goal-oriented mobile robot. If necessary, extra terms can be added to take into account other aspects; if avoiding narrow openings is a desired behavior, an extra term in the form of $\mu_4 \cdot \Delta(k_r, k_l)$ could be added. Opposite to this, if narrow openings are preferred, an additional term in the form of $\mu_4 \cdot \frac{1}{\Delta(k_r,k_l)}$ could be added. The function $\Delta(\cdot)$ is implemented as shown in Eq. (2-30).

$$\Delta(c_1, c_2) = \min \ \{|c_1 - c_2|, \ |c_1 - c_2 - n|, \ |c_1 - c_2 + n|\} \tag{2-30}$$

6. Velocity Similarly to the navigation method proposed by the Automata in Figure 3-1, the velocity here is computed as a function of the difference between the force applied by the operator and the repulsive force created by the environment as shown in Eq. (3-5).

2-4 Event-based Control

Event based control has gained noticeable attention as means to reduce the consequences of communication delays in networked control systems [21, 22, 23, 24], as it has a noticeable influence on its closed-loop performance. A general structure of the event-based control is shown in Figure 2-10 [25] and consists of:

- the plant with state vector $\mathbf{x}(t)$, the input vector $\mathbf{u}(t)$, the output vector $\mathbf{y}(t)$ and an unknown disturbance vector $\mathbf{d}(t)$,
- a control input generator which determines the event times at which the loop is closed, and
- a digital communication network which may cause communication delays τ

Definition 1. Event-based control [26]. Given a system in the form of Eq. (2-8) and under the assumption that the state variable x is measured without error, we define an interval $-\Delta < x < \Delta$ on which no control action is taken. Therefore the control action $u(t)$ is defined as:

$$u(t_{k_i}) \begin{cases} 0 & -\Delta < x(t_{k_i}) < \Delta \\ \neq 0 & |x(t_{k_i})| = \Delta \end{cases}$$

To provide insight on the event-based control, an example is provided as in [26]:

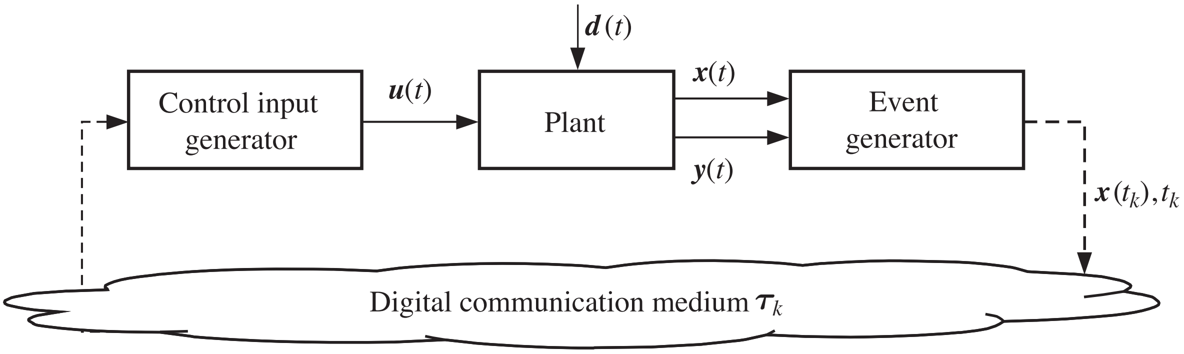


Figure 2-10: Event-based control loop

Example 2. Consider a system to be controlled in the form of:

$$dx = udt + dv \tag{2-31}$$

where $v(t)$ is the disturbance and $u(t)$ is the control signal. A region $-a < x < a$ is specified. No control action will be taken if the state is inside this region. Control actions are only taken at events t_k when $|x(t_k)| = a$. Therefore, the control input can be reduced for all the times on which the states fall within the control region.

To illustrate this, in Figure 2-11, a comparison between three different methods can be seen. The control objective is to maintain the system at the origin, i.e. $x = 0$.

The simulation is performed by approximating a continuous behavior by fast sampling. The states are shown in the upper plots and the control signals below them.

The first case is achieved by periodic sampling with Zero Order Hold (ZOH). With a sampling period h, the sampled system is then:

$$x(t+h) = x(t) + hu(t) + e(t) \tag{2-32}$$

The minimum variance over one sampling period is defined as:

$$V_{PZOH} = \frac{3+\sqrt{3}}{6}h \tag{2-33}$$

The second case (center of Figure 2-11) is achieved by using an Impulse Hold (IH). The control signal is applied when an event occurs, as defined in Eq. (2-34), where δ is the delta function.

$$u(t) = -x(t_k)\ \delta(t - t_k) \tag{2-34}$$

This control law implies that an impulse which resets the control to zero is applied. After the impulse, the closed-loop system is driven by $dx = dv$ and the variance grows linearly until the system reaches the border of the interval and a control action is needed. The variance over a sampling interval is then defined as:

$$V_{PIH} = \frac{1}{2}h \tag{2-35}$$

Finally, the third case in this example is the event based control approach. Here, the steady state variance is defined as:

$$V_{EBC} = \frac{h_E}{6} \tag{2-36}$$

where $h_E = a^2$ is the mean exit time.

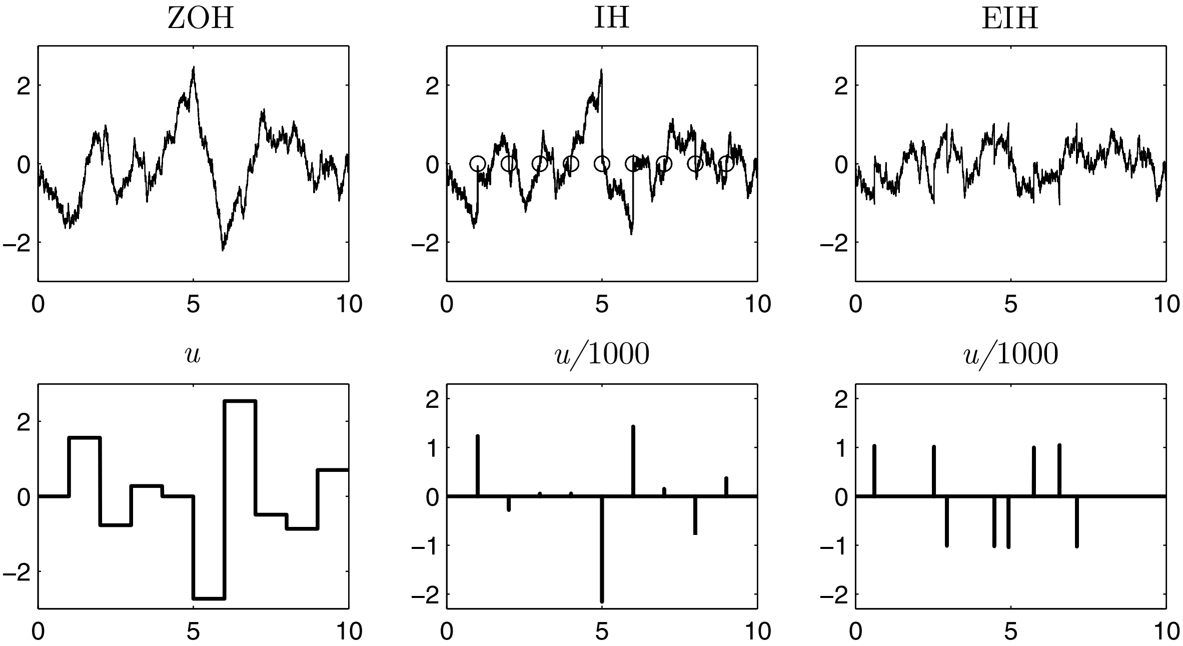


Figure 2-11: Simulation of an integrator with periodic sampling and ZOH (left), IH (center) and event based control with IH (right) [26]

From Figure 2-11 the following things should be noted:

- An event based controller gives a smaller variance than a controller with periodic sampling. The variances are thus related as $4.7h \ : \ 3h \ : \ a^2$,
- an event based controller acts as soon as the error is detected vs. when the sampling time h occurs,
- the process state remains within the bounds at all times.

Because the event based control approach yields promising results on the process variable with minimal control action, it has been extended to haptic data savings, this is covered in Chapter 3.

2-5 Dissipative and Passive systems

Passivity is a property of physical systems that can be roughly defined in terms of energy dissipation and transformation. A passive system is then defined as a system that cannot store more energy that is supplied, i.e. no internal energy generation only dissipation. Therefore, a system is stable if bounded inputs yield bounded outputs. Furthermore, the interconnection of two passive systems yields a passive system, therefore if the energy generated by one system is dissipated by the other one, the closed loop will be stable [27].

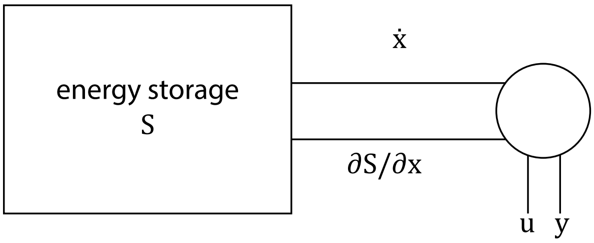


Figure 2-12: Power-preserving interconnection structure

Definition 3. Port-Hamiltonian system[28]. A lossless system as an energy storage port that interacts with the environment through a power-preserving structure as the one shown in Figure 2-12, yields differential algebraic equations (DAEs) of the form:

$$F(x)\begin{bmatrix}\dot{x}\\ y\end{bmatrix} = E(x)\begin{bmatrix}\frac{\partial S}{\partial x}(x)\\ -u\end{bmatrix}, \tag{2-37}$$

where $F(x),\ E(x)$ satisfy $F(x)E^T(x) + E(x)F^T(x) = 0$. By setting $F(x) = I_{2\times 2}$ and $E(x)$ as:

$$E(x) = \begin{bmatrix}J(x) & -g(x)\\ g^T(x) & 0\end{bmatrix},$$

yields the port-hamiltonian system in Eq. (2-38):

$$\begin{aligned}\dot{x} &= J(x)\frac{\partial S}{\partial x}(x) + g(x)\ u,\\ y &= g^T(x)\frac{\partial S}{\partial x}(x)\end{aligned} \tag{2-38}$$

where $x \in \mathbb{R}^n$, $S(x)$ is the total stored energy, $J(x)$ is a skew-symmetric matrix and $u,\ y \in \mathbb{R}^m$ are a conjugated input-output pair.

Due to skew symmetry of $J(x)$, the power-balance $\dot{S}$ reads:

$$\dot{S} = \frac{\partial^T S}{\partial x}(x)\dot{x}, \quad = \frac{\partial^T S}{\partial x}(x)\ J(x)\frac{\partial S}{\partial x}(x) + \frac{\partial^T S}{\partial x}(x)\ g(x)\ u, \quad = u^T\ y, \tag{2-39}$$

or equivalently, by integrating the power-balance Eq. (2-39) yields:

$$\underbrace{S\left(x(t_1)\right) - S\left(x(t_0)\right)}_{\text{Stored energy}} = \underbrace{\int_{t_0}^{t_1} u^T(t)\ y(t)\ dt}_{\text{Supplied energy}} \tag{2-40}$$

which is known as the energy balance equation for a lossless system.

Definition 4. Dissipativity [27] In the space $\mathbb{U}$ x $\mathbb{Y}$, a function is defined as $s:\ \mathbb{U}$ x $\mathbb{Y} \rightarrow \mathbb{R}$, called the supply rate. A space state system in the form of in Eq. (2-8), is said to be dissipative with respect to the supply rate s if there exists a function S: $\mathbb{R}^n \rightarrow \mathbb{R}^+$, called the storage function, such that for all x_0, all $t_1 \geq t_0$ and all input functions $u(\cdot)$

$$S\left(x(t_1)\right) \leq S\left(x(t_0)\right) + \int_{t_0}^{t_1} s\left(u(t), y(t)\right) dt, \tag{2-41}$$

where $x(t_1)$ is the state of Σ at time t_1 resulting from initial conditions $x(t_0) = x_0$ and input functions $u(\cdot)$. If Eq. (2-41) holds with equality, then Σ is lossless with respect to s.

Therefore an important choice of supply rate is $s(u, y) = u^T y$. Since the time-integral of power represents energy, the inequality (2-41) expresses the fact that the stored energy $S\left(x(t_1)\right)$ of Σ at any future time t_1 is at most equal to the sum of the stored energy $S\left(x(t_0)\right)$ at present time t_0 and the total externally supplied energy:

$$E(t) = \int_{t_0}^{t_1} s\left(u(t), y(t)\right) dt = \int_{t_0}^{t_1} u^T(t) y(t) dt \tag{2-42}$$

during the time interval $[t_0, t_1]$. This means that there is no internal creation of energy, only dissipation. Thus, as stated in [27], "a system that is dissipative with respect to the supply rate s cannot store more energy than it is supplied to; the input-output mapping $u \rightarrow y$ is *passive*".

Definition 5. Passivity [27] The system (2-8) is:

- *passive* if it is dissipative w.r.t. the supply rate $s = u^T y$;
- *strictly input passive* if $\exists\ \delta > 0$ such that it is dissipative w.r.t. $s = u^T y - \delta ||u||^2$;
- *strictly output passive* if $\exists\ \varepsilon > 0$ such that it is dissipative w.r.t. $s = u^T y - \varepsilon ||y||^2$;
- *conservative* if it lossless w.r.t. $s = u^T y$.

By considering a passive system in the shape of Eq. (2-8). Since the system is passive, it satisfies the *dissipation inequality* in Eq. (2-41). By differentiating (2-41) and assuming that $[t_0, t_1] = [0, t]$, yields:

$$\dot{S}(x) \leq s(u, h(x, u)),\ \forall\ x, y \tag{2-43}$$

This means that the rate of change of storage along the trajectories of the system will never exceed the rate of supply, i.e. no internal energy generation.

Stability of interconnected passive systems

Consider two state-space systems denoted by Σ_1 and Σ_2 on the form of Eq. (2-8), i.e.

$$\Sigma_i : \begin{cases} \dot{u}_i = f_i(x_i, u_i) \\ y_i = h(x_i, u_i) \end{cases} \tag{2-44}$$

with $x_i \in \mathbb{R}^{n_i}$, $u_i \in \mathbb{U}_i$, and $y_i \in \mathbb{Y}_i$. Suppose both systems are passive with storage functions $S_1(x_1)$ and $S_2(x_2)$ respectively, i.e.

$$S_1\left(x_1(t_1)\right) \le S_1\left(x_1(t_0)\right) + \int_{t_0}^{t_1} s\left(u_1(t), y_1(t)\right) dt, \tag{2-45}$$

$$S_2\left(x_2(t_1)\right) \le S_2\left(x_2(t_0)\right) + \int_{t_0}^{t_1} s\left(u_2(t), y_2(t)\right) dt, \tag{2-46}$$

And applying a feedback interconnection as the one shown in Figure 2-13, and with the feedback constraints $u_1 = e - y_2$ and $u_2 = y_1$. From here, it can be deduced that $u_1^T y_1 + u_2^T y_2 = e^T y_1$ and the addition of both dissipation inequalities (2-45) and (2-46) results in:

$$S_1\left(x_1(t_2)\right) + S_2\left(x_2(t_2)\right) \le S_1\left(x_1(t_1)\right) + S_2\left(x_1(t_1)\right) + \int_{t_1}^{t_2} \left(e^T(t), y_1(t)\right) dt$$

Hence the closed-loop system with input e and output y_1 is passive with storage function as shown in Eq. (2-47). With this, stability can be determined for a complex system if each of the subsystems is passive [27, 29].

$$S(x_1, x_2) = S_1(x_1) + S_2(x_2), \; (x_1, x_2) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \tag{2-47}$$

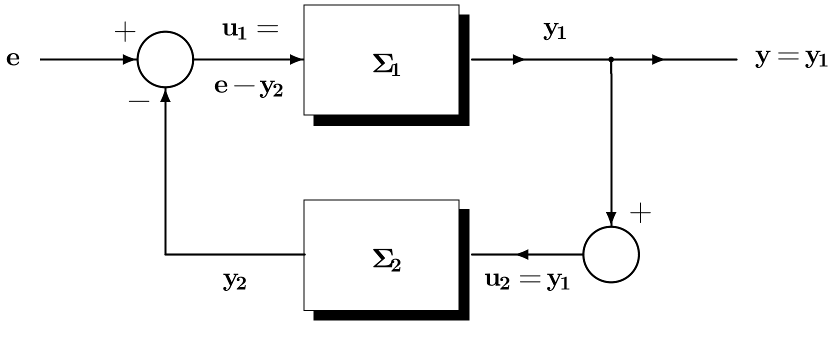

e
+
−
$\mathbf{u_1} =$
$\mathbf{e} - \mathbf{y_2}$
Σ_1
$\mathbf{y_1}$
$\mathbf{y} = \mathbf{y_1}$
+
Σ_2
$\mathbf{y_2}$
$\mathbf{u_2} = \mathbf{y_1}$

Figure 2-13: Feedback interconnection between two systems

2-6 Wave transformation

In order to preserve stability of a system under the presence of delays, a method first introduced in [30] and continued in [31] known as *scattering* is suggested. With this, data is transformed from the mechanical domain to the waves domain, i.e. packaged in waves, to allow the data to be sent over the communication channel with a delay τ and unwrapped on the other end preserving passivity of the system.

In the Mechanical domain, the effort e and flow f variables are the velocity $\dot{x}$ and force F. Therefore, the power function, defined as $P = e \cdot f$ for the mechanical domain is:

$$P_{\text{mech}} = \dot{x} \cdot F$$

In order to minimize the effect of distortions caused by delays on the communication channel, the following domain transformation is suggested by [30]:

$$\begin{aligned} u &= \dot{x}\sqrt{2b} - v \\ F &= \dot{x}\ b - v\sqrt{2b} \end{aligned} \tag{2-48}$$

By rearranging Eq. (2-48) and solving for the wave-variables u and v we have:

$$\begin{aligned} u &= \frac{b\ \dot{x} + F}{\sqrt{2b}} \\ v &= \frac{b\ \dot{x} - F}{\sqrt{2b}} \end{aligned} \tag{2-49}$$

From Eq. (2-49), the power function becomes:

$$P_{\text{wave}} = \frac{1}{2}u^T u - \frac{1}{2}v^T v, \tag{2-50}$$

from which it can be seen that the power function now becomes a summation, making it less sensitive to the distortion caused by the communication channel.

2-7 Sketch of the solution

The goal of this project is to compare the performance of a remotely-controlled robot navigation system by implementing two collision avoidance algorithms paired with force feedback systems. The virtual forces are meant to provide extra information to the operator through a haptic device. The testing setup is based on a non-linear robot as the one described in Section 2-2 and shown in Eq. (2-8). As the system is meant to be used in a Teleoperation scheme as shown in Figure 2-1, the performance of the system is dependent on the communication channel, therefore a reduction on the information transmitted is explored through the use of event-based control. Finally the stability of the whole telepresence system is going to be preserved through passivity as introduced in Section 2-5.

Additionally, the implementation of the Vector Field Histogram as a feedback force generator is going to be investigated as a mean to reduce the number of algorithms that need to be executed to achieve the collision avoidance with haptic feedback system.

Chapter 3

Solutions

In this chapter, the different methods compared in this project are described. Because of the characteristics of the project, it can be divided into two main areas of focus: *Collision avoidance and Force feedback* methods covered in Section 3-1 and *Data compression and delay management* methods are covered in Section 3-2. The first section will describe the methods available to translate objects into haptic data and how this can be used for collision-free navigation. The second section describes the methods to compress haptic data without losing the transparency of the system.

3-1 Collision avoidance and Force feedback

3-1-1 Artificial force field

As described in Section 2-3-1, the first approach to follow is the Parametric Risk Field (PRF). The parameters used are:

d_{min}	0.01 m
a_{max}	2 m/s^2
r_{pz}	0.25 m

Table 3-1: Parameters used for the PRF algorithm

This guarantees that when an obstacle is at a distance $d_{obs} \leq d_{min} = 10$ cm, the repulsive force will be on its maximum value.

Collision Avoidance and attractive force

As discussed in Section 2-3-1, the PRF algorithm computes a series of obstacle vectors, contain information such as the shortest distance to an obstacle and the risk area, and its direction.

With this information and the goal coordinates, the collision detection algorithm can be implemented as an automata to determine the direction of the feedback forces. The output of the following algorithm is a direction on which it is safer to move. A user configurable parameter, $\alpha \in [0, 1]$, should be set in order to determine the behavior of the system.

The algorithm proposed by [32] is as follows:

1. Get the distances detected by the sensors (PRF),
2. transform these values into the robot frame by the use of a rotation matrix shown in Eq. (3-1). Here, the robot is considered a point in space and therefore, no translation is performed, hence $x = 0,\ y = 0$ and $\varphi = \varphi_{s,i}$ is the angle of the sensor s_i (PRF),

$$R(\varphi) = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & x \\ \sin(\varphi) & \cos(\varphi) & y \\ 0 & 0 & 1 \end{bmatrix} \tag{3-1}$$

3. transform into the world frame values by using Eq. (3-1) with: $\varphi =$ robot's orientation with respect to the Z axis, $x = x_{robot}$ and $y = y_{robot}$,
4. compute the distance from each of the detected obstacles to the Robot Center Point (RCP) and multiply it by a gain-matrix $\mathbb{K}_s$ with size [1 x n], where $n = i$ is the number of sensors. The purpose for this matrix is to increase the weight of front-facing sensors and diminish the effect of the side-sensors for the collision avoidance algorithm.
5. with the minimum distance to the obstacles, the automata model shown in Figure 3-1 is followed. The guard conditions are based on the distance,
6. based on the current machine state the robot is located, the direction for the feedback forces are computed and sent over the communication channel. Back to step 1.

Machine state description

As shown in Figure 3-1, the active state depends on the distance from the robot to the obstacle and the robot to the goal. Each one of the states has a different approach for the robot navigation and they are explained below:

no obstacle This state, as its name suggests, is triggered when there is no obstacle detected within a user-defined obstacle distance $d_{obs} = 0.7m$. This distance is always smaller than the sensor's max distance. The direction γ_c for the attractive force is computed by the difference of the robot's current position and the goals, the go-to-goal vector $\hat{\mathbb{U}}_{GTG}$ as shown in Eq. (3-2); the allowed speed is the maximum achievable speed $v_r = v_{max}$ as there are no repulsive forces acting on the robot.

$$\hat{\mathbb{U}}_{GTG} = \begin{bmatrix} g_x - x_o \\ g_y - y_o \end{bmatrix} \tag{3-2}$$

$$\gamma_c = \text{atan}\ \left(\hat{\mathbb{U}}_{GTG}\right) = \text{atan}\left(\frac{\mathrm{r}_y - \mathrm{g}_y}{\mathrm{r}_x - \mathrm{g}_x}\right)$$

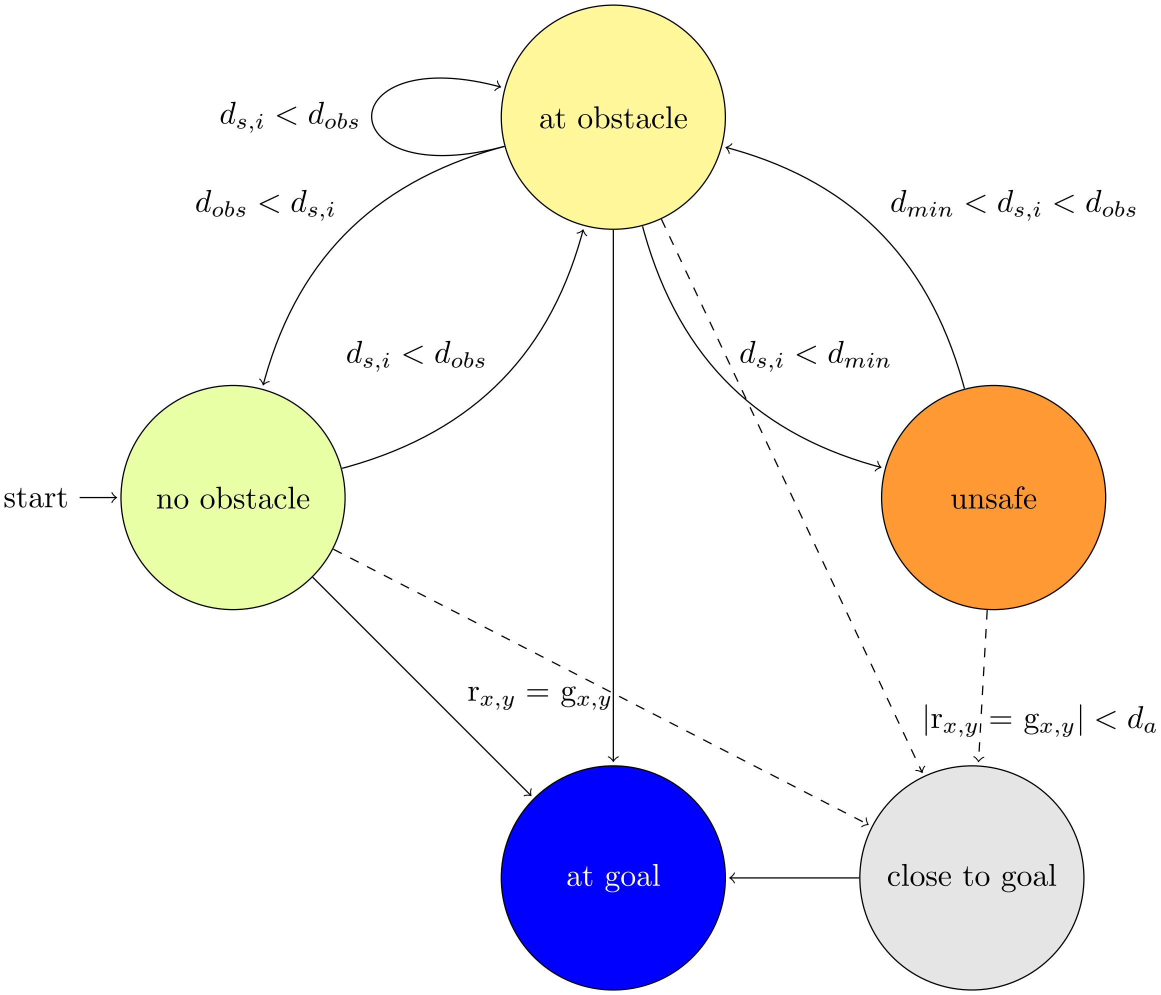


Figure 3-1: Automata: Shared-control scheme

at obstacle the condition to access this state is that the sensed distance $d_{s,i}$ is smaller than the obstacle distance and greater than the safe distance d_{min}, i.e. $d_{min} < d_{s,i} < d_{obs}$. The direction γ_c in this state is computed by blending the obstacle avoidance vector $\hat{\mathbb{U}}_{AO}$ and the go-to-goal vector $\hat{\mathbb{U}}_{GTG}$ (Eq. (3-3)):

$$\gamma_c = \text{atan}\left(\hat{\mathbb{U}}_{AO\_GTG}\right) = (\alpha \cdot \hat{\mathbb{U}}_{GTG}) + (1-\alpha) \cdot \hat{\mathbb{U}}_{AO} \tag{3-3}$$

where $\alpha \in [0,1]$ is the trade-off value between a goal-seeking ($\alpha = 1$) or a collision avoidance direction ($\alpha = 0$) and $\hat{\mathbb{U}}_{AO}$ is defined as in Eq. (3-4):

$$\hat{\mathbb{U}}_{AO} = \sum_{i=1}^{n} \left(d_{wf_{x,y}}(i) - r_{x,y}\right) \cdot \mathbb{K}_s \tag{3-4}$$

where n is the number of sensors, $d_{wf_{x,y}}(i)$ represents the coordinates of the obstacle i in the world frame, $r_{x,y}$ the robot's coordinates and the sensor-gain matrix $\mathbb{K}_s$.

On this state, the maximum velocity is determined by the repulsive forces created by the PRF algorithm. Therefore, the velocity is a function of the difference between the operator's force on the controller F_m and the repulsive force F_s, this is:

$$v_r = f\left(\text{F}_m - \text{F}_s\right) \tag{3-5}$$

unsafe this state is activated when the condition $d_{s,i} < d_{min}$ is fulfilled, representing an imminent danger to the robot and its surroundings. The direction γ_c for this state is the opposite direction of the current robot's orientation, i.e. $\gamma_c = \varphi_R + \pi$. Once this state is reached, the algorithm takes over the control and takes action to prevent collision. Once this has been fulfilled (back to operator state) it grants the control back to the operator. To prevent the collision, this state will drive the robot until the system leaves this state, i.e. reaching the *at obstacle* state.

close to goal is a shared-control state. This state is activated when the distance between the robot and the goal is within a threshold d_a, i.e. when the condition $|\text{robot}_{x,y} - \text{goal}_{x,y}| <= d_a$ is met. On this state, the maximum speed of the robot $v_r = \iota\ v_{max}$ with $\iota \in [0,1]$ is limited to increase the precision.

at goal the guard condition for this state is that the robot has reached the goal, i.e. that $\text{r}_{x,y} = \text{g}_{x,y}$.

3-1-2 Vector field histogram

In Section 2-3-2, the method to create a collision avoidance algorithm is explained. To implement it on these tests, the window size $w_s = 3.5m$. A minimum distance of 0.1 m was determined in order to prevent collisions; this means that the robot's size has been extended by 10 cm in all directions, guaranteeing enough space to turn and choose a new path without collisions. The thresholds $\tau_{high} = 15$ and $\tau_{low} = 5$ were assigned and a value of $s_{max} = 10$ determines when an opening is considered wide or narrow.

Force feedback

As discussed in Section 2-3-2, the Vector Field Histogram + (VFH+) algorithm is an algorithm intended for navigation. To take advantage of the information collected by the collision avoidance, the following operations to generate the repulsive force are proposed:

1. In order to generate the repulsive force, the output of the polar histogram can be used as it computes a force magnitude based on the certainty $c_{i,j}$ and the distance $d_{i,j}$.
2. In order to take into account the direction of movement, the sectors k of interest are ±30° of the robot's orientation,
3. the maximum magnitude in the sectors $k \pm 30°$ is used as the repulsive force:

$$F_s = \max\ (m_\kappa) \cdot \psi \text{ where } \kappa \in [k - 30°, k + 30°] \tag{3-6}$$

where ψ is a user-defined scaling factor for the magnitude.

3-2 Intelligent Data Management

3-2-1 Delay management

In order to preserve stability of the system under constant delay τ, the passivity method introduced in Chapter 2 is used. To guarantee stability, the proposed approach is passivity preservation through the *wave transformation.*

Wave transformation

To be used in a teleoperation scheme, the wave transform introduced in Section 2-6 will be used with the following changes: a forward moving wave u_m (master-to-slave communication) and a backward moving wave v_s (slave-to-master communication) are defined as follows:

$$\begin{aligned} u_m &= \frac{F_m + b\ \dot{x}_m}{\sqrt{2b}} \qquad u_s = \frac{F_s + b\ \dot{x}_s}{\sqrt{2b}} \\ v_m &= \frac{b\ \dot{x}_m - F_m}{\sqrt{2b}} \qquad v_s = \frac{b\ \dot{x}_s - F_s}{\sqrt{2b}} \end{aligned} \tag{3-7}$$

where $\dot{x}_m$ represents the Human-System Interface (HSI) velocity, F_m is the force applied by the environment to the HSI, $\dot{x}_s$ is the desired speed (coming from the operator or the automata), F_s is the force generated by the environment on the slave side, and b is the characteristic impedance, a strictly positive parameter; the consequence that the characteristic virtual impedance has on the closed-loop system, the higher this characteristic impedance becomes, the faster the poles are regardless of the delay [30, 31].

The scattering (wave) variables u_m (forward path) and v_s (backward path) are transmitted over the communication channel and arrive at the receiver with a time delay τ:

$$u_s = u_m(t-\tau) \quad v_m = v_s(t-\tau) \tag{3-8}$$

The wave transformation scheme proposed is shown in Figure 3-2. It is important to note that when a wave transformation scheme is used, wave reflections may occur at both local and remote sites. In the context of teleoperation, these waves are not desired as they would corrupt the information and cause oscillatory behavior, creating an extra workload on the operator in order to compensate for them. To avoid these reflections, an impedance matching term $I = b$ is applied [30].

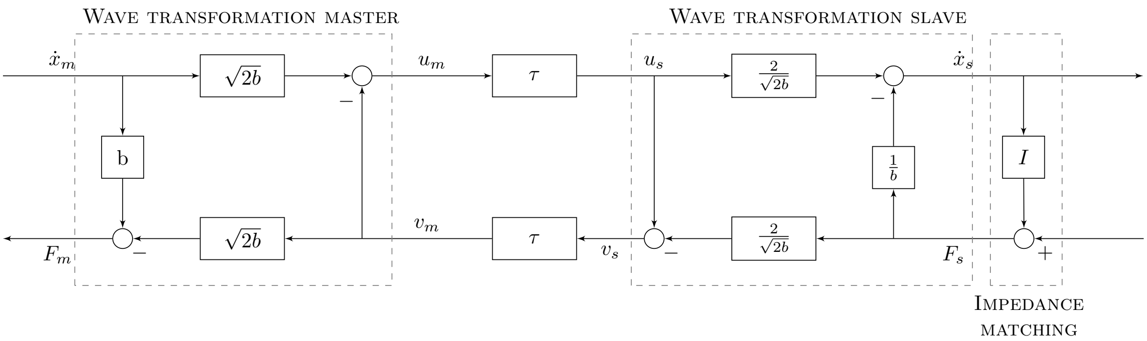
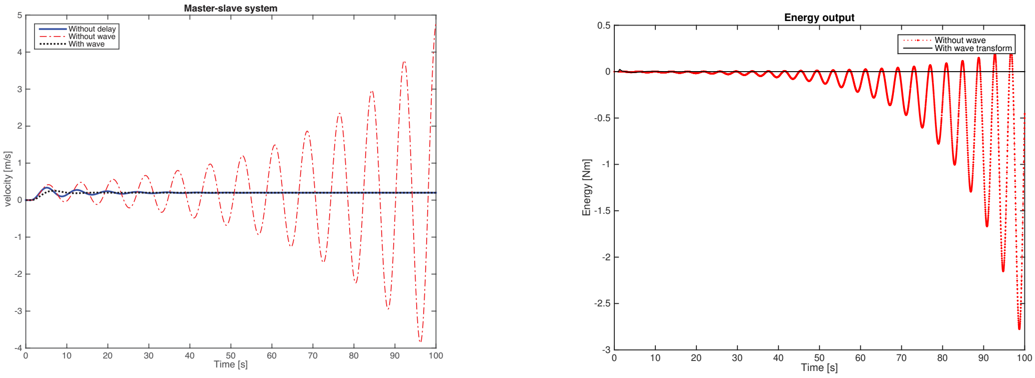


Figure 3-2: Wave transformation scheme for teleoperation [31].

Example 6. In Figure 3-3a we can see the results of applying the wave transformation in a master-slave tracking system. The test scheme is the same as the one implemented in [9] with the master controller is modeled as a mass-spring-damper system with a transfer function $G_m(s)$ and the slave $G_s(s)$ as follows:

$$G_m(s) = \frac{1}{0.0175s^2 + 2.2s + 3.5} \quad G_s(s) \quad = \frac{1}{1.45s} \tag{3-9}$$



(a) Slave position with a delay of $\tau = 200$ ms.

(b) Energy output with $\tau = 200$ ms

Figure 3-3: Master-slave tracking with delay.

3-2-2 Data compression

In the following section, different techniques to compress the haptic data without losing transparency are presented. These techniques work on an event-based control approach. In order to set the state limits described in Chapter 2, a minimum perceivable threshold (just-noticeable difference (JND)) on the haptic force change is used. This follows the research performed b E. H. Weber [33], known as Weber's law, which states that there exists a discrimination threshold for haptic stimuli. Based on this approach the following techniques are studied.

Deadband

Hirche et al. [34] propose a deadband algorithm based in event-based control as described in Chapter 2. This method has two subdivisions, a *constant* and a *relative* deadband approach. With deadband control, only relevant packages are sent over the network. In order to determine which packages are relevant, a deadband is implemented by comparing the last sent packet with the current one, if the difference between the current package and the previous one falls within the deadband the packet will not be transmitted, otherwise it is transmitted and used as a reference for the following haptic data packet.

In their research, [34], the authors show that haptic data "behaves approximately to Weber's law , i.e. linearly increases with the stimulus intensity". Therefore the deadband control works by comparing the value $u_{m,s}(t-1)$ with the current value $u_{m,s}(t)$ for $t-1 > t$. If the absolute difference is below this deadband threshold value Δ, the value will not be transmitted. The event indicator $\Omega(t)$ is then formulated as:

$$\Omega(t) = \begin{cases} 1 & \text{if } |u_{m,s}(t-1) - u_{m,s}(t)| \geq \Delta \\ 0 & \text{otherwise} \end{cases}$$

where 1 indicates a packet transmission and 0 otherwise.

Constant deadband. This states that the deadband value has a constant width (hysteresis). If the most recent transmitted value is close to the origin i.e. $u_{m,s}(t-1) < \Delta$, the sign could change, therefore as soon as it changes sign it must be transmitted. To consider this exception, the deadband is defined as follows:

$$|u_{m,s}(t)| \in \begin{cases} [0, |u_{m,s}(t-1)| + \Delta) & \text{if } |u_{m,s}(t-1)| < \Delta \\ [|u_{m,s}(t-1)| \pm \Delta] & \text{if } |u_{m,s}(t-1)| \geq \Delta \end{cases}$$

with this, the sign condition

$$u_{m,s}(t)\ u_{m,s}(t-1) > 0 \tag{3-10}$$

between transmitted and current values is guaranteed.

Relative deadband. The width for this deadband grows linearly with the magnitude of the value of $u_{m,s}(t-1)$. The proportional value ϵ is a value which should be tuned for a desired performance on the haptic interface. With this, the relative deadband value is defined by:

$$\Delta_{u_{m,s}(t-1)} = \epsilon \cdot |u_{m,s}(t-1)| \tag{3-11}$$

If the signal $u_{m,s}(t-1)$ is close to the origin, the deadband becomes infinitely small, therefore the authors suggest a lower bound $\Delta \geq \Delta_{\min}$. In order to guarantee the sign consistency argument (3-10) the relative deadband is implicitly defined by:

$$|u_{m,s}(t)| \in \begin{cases} [0, |u_{m,s}(t-1)| + \Delta_{\min}) & \text{if } |u_{m,s}(t-1)| < \Delta \\ [|u_{m,s}(t-1)| \pm \Delta_{u_{m,s}(t-1)}] & \text{if } |u_{m,s}(t-1)| \geq \Delta \end{cases}$$

Limitations In teleoperation applications, it is pretty common to encounter haptic devices with two or more Degrees of Freedom (DoF), usually composed of two or more cartesian components of the current velocity or force. The deadband approach discussed in this section only takes into account systems with one DoF; applying it to every DoF is a straightforward extension, however inefficient [6]. The reason behind this is described by the authors: "if random movements with identically distributed directions and magnitudes of forces and velocities are examined, the com- ponent with the lowest magnitude and therefore the smallest deadband is mostly responsible for packet generation." Therefore, the number of DoFs is inversely proportional to the performance of this approach. An extension to multiple DoF can be found in Section B-1

Stability

Each time a packet is not sent, empty packet instances can be found at the receiver side. To cope with this, the emptied data needs to be estimated. Under the assumption that the most recent data u_r arrived at the time $t^* = t - 1 + \tau$ where τ is a time delay, the reconstruction operator can be denoted by:

$$u_r(t) = \begin{cases} u_{m,s}(t-\tau) & \text{if } \Omega(t-\tau) = 1 \\ \zeta(u_r(t^*), t) & \text{otherwise} \end{cases} \tag{3-12}$$

where $\zeta(\cdot)$ denotes a reconstruction algorithm.

Deadband Passivity

In order to preserve passivity of the system, an "Energy Supervised Data Reconstruction" algorithm is implemented. The passivity observer is proposed as follows:

$$\zeta(t) = \begin{cases} \zeta_{np}(t) & \text{if } E(t) \geq 0 \\ \zeta_p(t) & \text{otherwise} \end{cases} \tag{3-13}$$

where ζ_{np} and ζ_p are a standard reconstruction strategy and a strictly passive strategy, respectively. Two strategies derive from here, a Hold Last Sample (HLS) or *zero order hold*, and a modified HLS (strictly passive), presented in [34] as follows:

The communications between master and slave (u_m and v_s respectively), are transmitted over the network and arrive with a time delay τ as shown in Eq. (3-14):

$$u_s(t) = u_m(t-\tau); \quad v_m(t) = v_s(t-\tau) \tag{3-14}$$

If no data packet has been received at the time t, corresponding to $\Omega(t-\tau) = 0$, then the value at the sender $u_m(t-\tau)$, and therefore $u_s(t)$, must lie within the deadband interval. The modified HLS algorithm:

$$\zeta(u_s(t^*), t) = u_s(t^*) - \text{sign } \{u_s(t^*)\} \cdot \Delta \tag{3-15}$$

reconstructs the missing data. As discussed before, it has a passivity preserving character by considering the energy balance Eq. (3-18). By computing the wave value as shown in Eq. (3-13), the reconstruction algorithm results in Eq. (3-16):

$$u_s^2(t) = \begin{cases} u_m^2(t-\tau) & \text{if } \Omega(t-\tau) = 1 \\ (|u_s(t^*)| - \Delta)^2 & \text{otherwise} \end{cases} \tag{3-16}$$

With this transformation, the power balance of the communication system is defined by:

$$P_{c,in} = \frac{1}{2}\left(u_m^2 - u_s^2\right) - \frac{1}{2}\left(v_s^2 - v_m^2\right) \tag{3-17}$$

Passivity of this system is then guaranteed if

$$E_{cf,in}(t) = \int_0^t \left(u_m^2 - u_s^2\right) d\tau \geq 0 \tag{3-18}$$

holds for the forward path and equivalently for the backward path. Therefore, taking into account the output wave power in (Eq. (3-12)) and the reconstruction algorithm in (3-15), yields:

$$u_s^2(t) = \begin{cases} u_m^2(t-\tau) & \text{if } \Omega(t-\tau) = 1 \\ (|u_s(t^*)| - \Delta)^2 & \text{otherwise} \end{cases}$$

This algorithm is interpreted by the authors as “a worst case estimation of the untransmitted data corresponding to a minimal wave input energy assumption”, i.e. dissipating energy in order to preserve passivity and therefore, stability.

3-2-3 Force Discrimination

The following table, obtained from [35], presents the force discrimination criteria for haptic related activities. From here the deadband thresholds can be obtained.

Physical property	**JND**	**Experimental conditions**
Force	$7 \pm 1\%$	arm/forearm
	ca. 15%	arm, static force
	ca. 10%	arm/forearm [36, 37]
Movement	$8 \pm 4\%$	arm/forearm
Position	$8 \pm 2\%$	arm/forearm
Stiffness	$23 \pm 3\%$	arm/forearm
	8%	pinch-fingers

Table 3-2: Perceptual discrimination thresholds JNDs for haptic-related properties

From the previous table, it can be seen that a Force discrimination of $7\% - 15\%$ has been detected. Based on these findings, the constant deadband will make use of these discrimination rules by using a discrimination threshold of 10%.

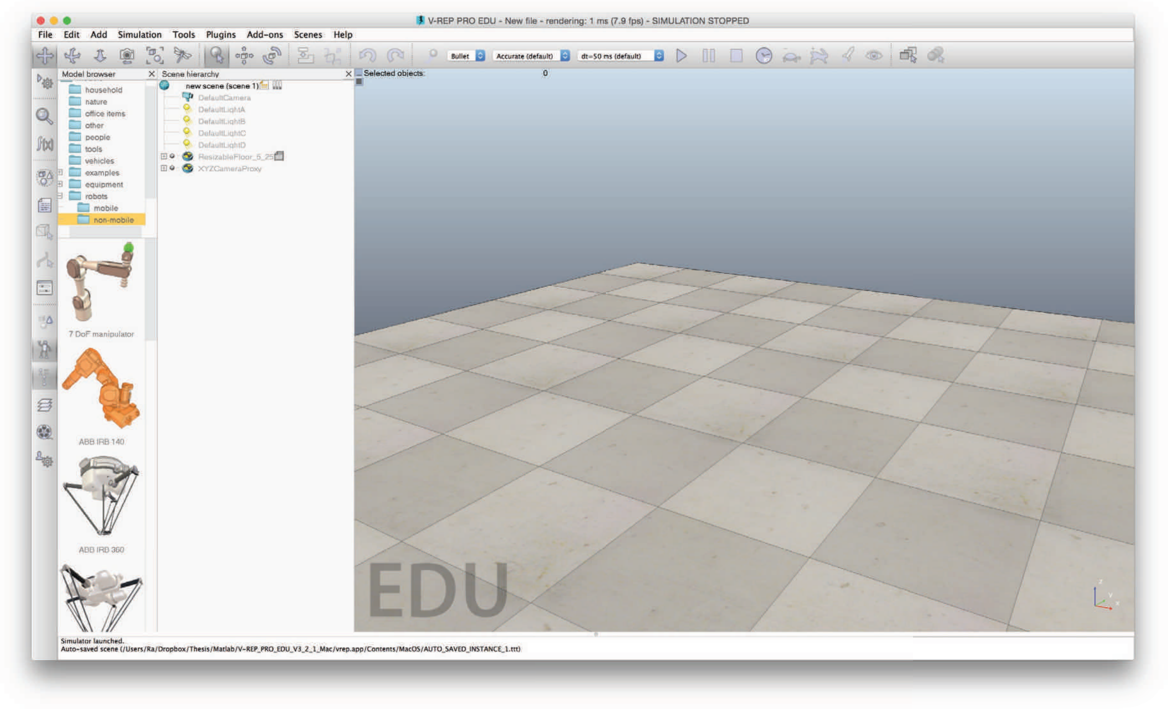
Chapter 4

Experimental Setup

4-1 Simulations

4-1-1 Setup

The simulator used is V-REP [38], a computer simulator developed to solve the problem derived by the integration of actuation, sensing and control, which makes robotics systems powerful, but complicates their simulation. This software is a versatile, scalable, yet powerful general-purpose robot simulation framework. The main interface is shown in Figure 4-1.

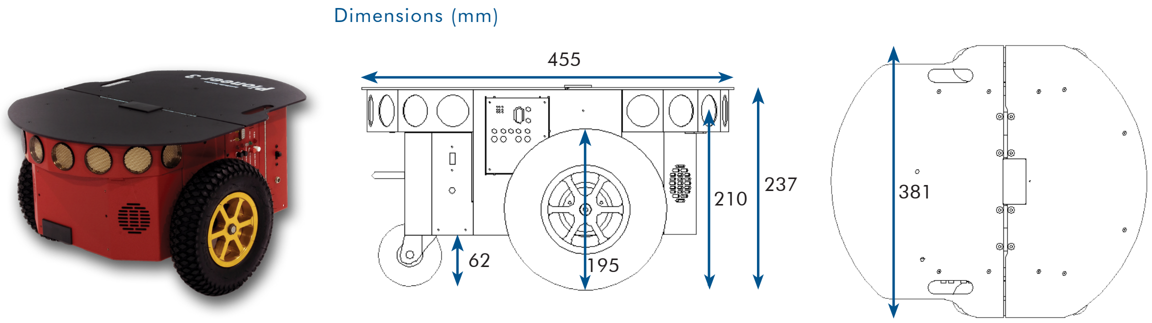


```
Simulator launched.
Auto-saved scene (/Users/Ra/Dropbox/Thesis/Matlab/V-REP_PRO_EDU_V3_2_1_Mac/vrep.app/Contents/MacOS/AUTO_SAVED_INSTANCE_1.ttt)
```

Figure 4-1: V-REP: a robot simulation framework.

4-1-2 Robot

The robot chosen for this experiments is the Pioneer P3-DX [39]. This robot is chosen because of its wide capabilities and software interaction. A 3D model representation of this robot comes with V-REP, allowing a more intuitive interaction with the simulator. The schematics of the P3-DX are shown in Figure 4-2b.



(a) Pioneer P3-DX

(b) Pioneer P3-DX schematics

Figure 4-2: Pioneer P3-DX

The Pioneer P3-DX's technical specifications are listed in Table 4-1:

Body	1.6 mm aluminum (powder-coated)
Tires	Foam-filled rubber
Robot Weight	9 kg
Operating Payload	17 kg
Turn Radius	0 cm
Swing Radius	26.7 cm
Max. Forward/Backward Speed	1.2 m/s
Rotation Speed	300°/s
Sensors	16 ultrasonic sensors

Table 4-1: P3-DX specifications

Other relevant parameters of this robot are shown in Table 4-2.

Wheel radius R	19.5/2 cm
Axis length d	38.1 cm

Table 4-2: P3-DX parameters

4-1-3 Maps

There are three maps that will be used for the simulations. They differ from one another based on the type of objects; obstacles besides walls should be considered for the collision avoidance.

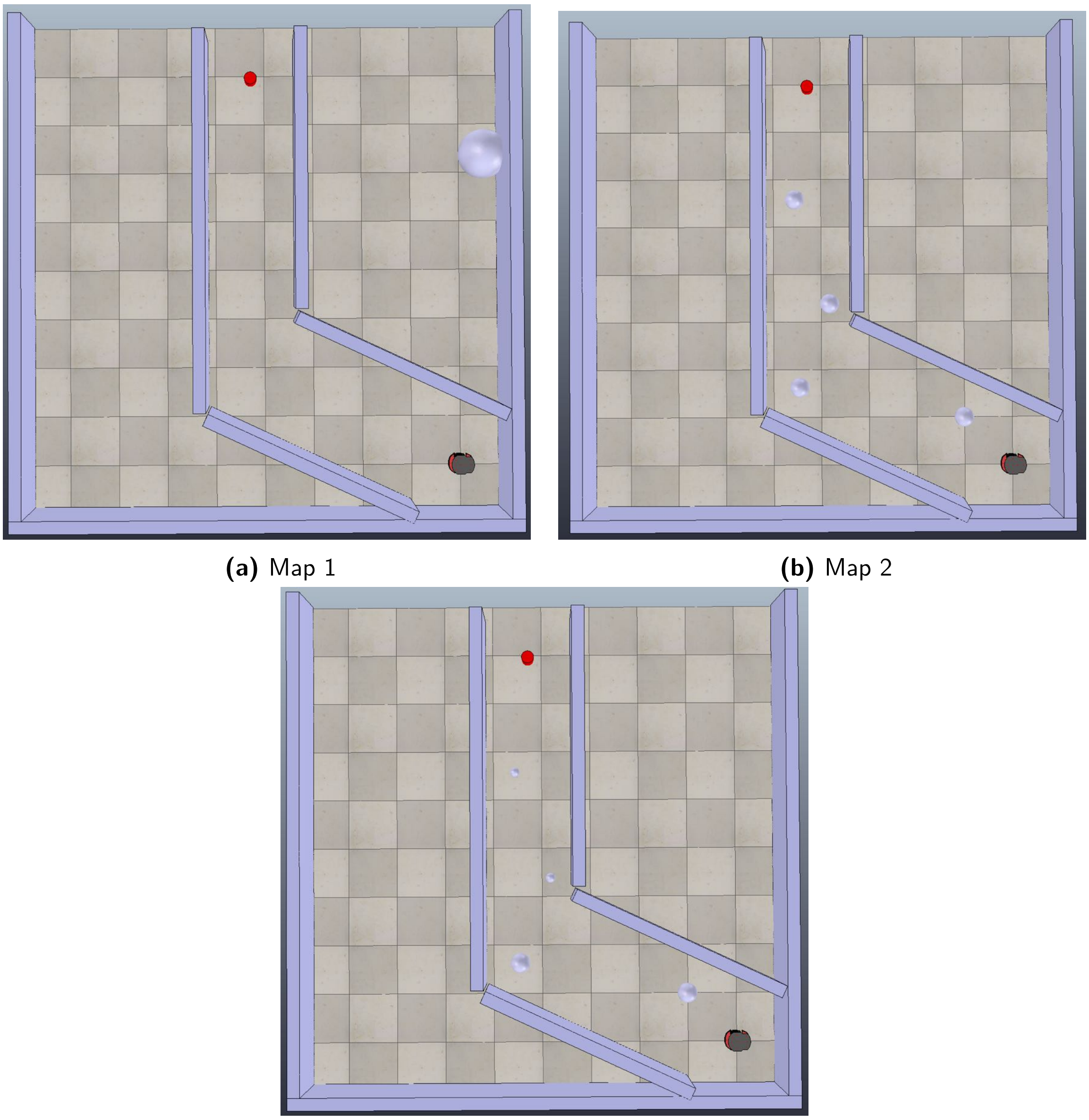
In Figure 4-3a we can see the first map. The starting point is illustrated by the robot's position and the goal is represented by the red cylinder. The size is approximately 100 m^2 with a trajectory of 12 meters approximately. On the first map, no obstacles are on the robot's path, only walls determining the path-to-follow are used.

The maps' complexity increases gradually for the second and third map by adding obstacles in the shape of a sphere with radius of 0.4 meters, as shown in Figure 4-3b. This represents the second simulated environment, the size is the same as in Figure 4-3a. Finally, the third simulation environment is shown in Figure 4-3c. In order to increase the complexity of the environment, the obstacle size was decreased by half, i.e. spheres with 0.2 and 0.4 meters of radius are added in the robot's path.

4-1-4 Controller

Autopilot

Several studies have been performed to model a human in a teleoperated-robot system loop as the one shown in Figure 4-4 [40]. First in the automotive domain [41] on which laterial, longitudinal, path-following, headway control and obstacle avoidance tasks are studied. Furthermore, physical characteristics unique to human are described; example driver models are generated to predict performance.



(a) Map 1

(b) Map 2

(c) Map 3

Experiments have shown that time delays greater than 40 ms produce degradations in performance for simple positional control tasks [42]. Furthermore, reaction-time experiments show that visual response times are about 180 ms. Comparable auditory and tactile response times are about 140 ms [43, 44]. Therefore, the presence of these delays generate a need for a shared-control scheme on which the operator shares the control with the collision avoidance algorithm.

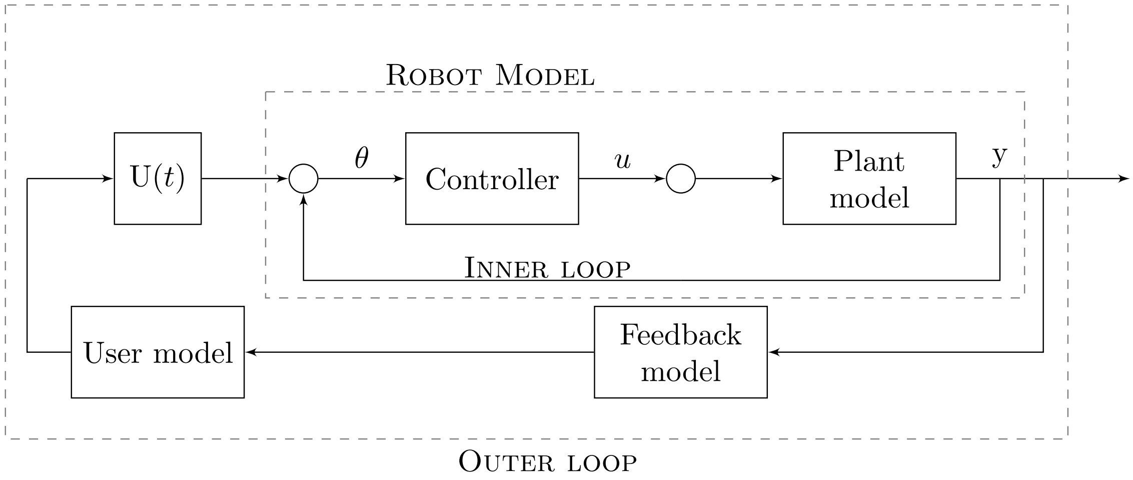


Figure 4-4: Elements in teleoperation. The robot processes run in the inner loop, while the operator provides control for the outer loop.

Even though the complexity of a human operator makes it hard to model it precisely, low order models are often sufficient for several control tasks, e.g. navigation [45], in the form of discrete-time ARX models for closed-loop control tasks. Good performance was obtained for both models for a tracking path model, however no testing has been performed for collision avoidance.

In the paper written by Arif and Innoka [46] experiments have been done to study the human capability to perform tasks by learning iteratively. It is concluded that the human operator performs the repetitive task by modifying his control action using the perception of error and error rate, in each iteration. Therefore, it becomes reasonable to implement an autopilot through the implementation of a feedback control scheme. The chosen controller therefore is a PD controller based on the orientation error $e_k = \gamma - \hat{\varphi}_R$, where γ is the output of the collision avoidance systems described in Section 3-1, and $\hat{\varphi}_R$ is an estimation of the current robot's orientation.

From Eq. (4-2), we have the desired angular velocity as a function of the error. This value becomes an input for the P3-DX block and through Eq. (4-1) and the values in Table 4-2, the angular velocity for each of the wheels of the differential drive robot is computed. This creates a new orientation $\hat{\varphi}_R$. The block diagrams for such a structure are shown in Figure 4-5.

$$\omega_r = \frac{2v + \omega d}{2R} \qquad \omega_l = \frac{2v - \omega d}{2R} \tag{4-1}$$

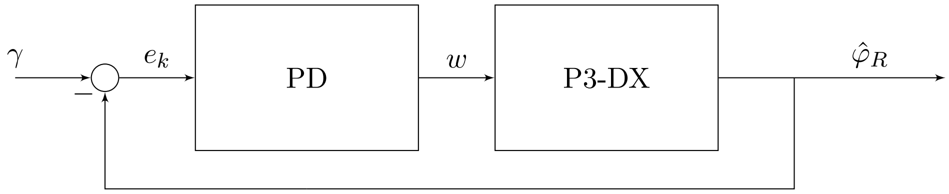


Figure 4-5: Autopilot PD control

$$\begin{aligned} e_k &= \gamma - \hat{\varphi}_R \\ \dot{\varphi} &= \omega \\ \omega &= K_P \cdot e_k + K_D \cdot \dot{e_k} \end{aligned} \tag{4-2}$$

where K_P, and K_D are the PD gains.

The values for each of the PD gains were based on the Ziegler-Nichols ultimate sensitivity method [47]. Several revisions have been made to this method [48, 49]. However, these revised methods are intended for auto-tuning controllers, which lie out of the scope of this project. The Ziegler-Nichols parameters are shown in Table 4-3.

Type	K_P	T$_I$	T$_D$
P	0.5 k_c		
PI	0.4 k_c	0.8 T$_c$	
PID	0.6 k_c	0.5 T$_c$	0.125 T$_c$

Table 4-3: Ziegler-Nichols ultimate sensitivity method

where T$_c$ is the ultimate period and k_c is the critical gain. After online tuning the PD controller using the *non-parametric tuning methods* [50], the resulting gains are shown in Eq. (4-3), the system's behavior under a step input is shown in Figure 4-6. It can immediately be seen that the controller performance shows undershoot, which is desired as minimal overshoot could represent a risk situation; studies in human behavior while steering show this kind of behavior for human operators [51, 52]. The controller yields the values shown in Table 4-4.

Rise time	1.6 [s]
Overshoot	0%
Settling time	2.8 [s]
Steady-state error	0.01 [rads]

Table 4-4: Transient response with PD control.

$$\begin{aligned} K_P &= 3 \\ K_D &= 0.2 \end{aligned} \tag{4-3}$$

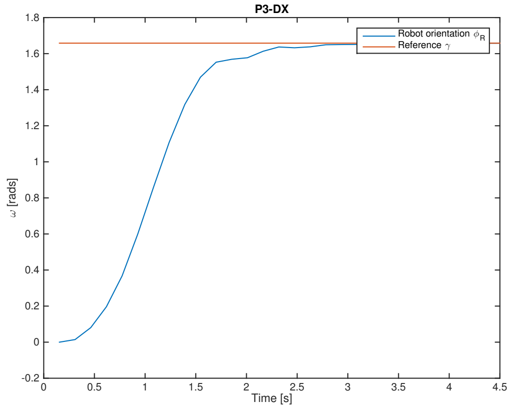


Figure 4-6: Step response with PD control

Force to velocity mapping

Due to the properties of the wave variables and to implement the force feedback in the setup, the scheme used in Figure 4-7 is implemented. Here, as it can be seen, a simulated steering wheel is used to map the feedback force F_m to a velocity $\dot{x}_{F_m}$ (similar to the implementation in [53]), which is then added to the velocity v_{ref} provided from the Automata, which results in the velocity $\dot{x}_m$, which is sent over the communication channel.

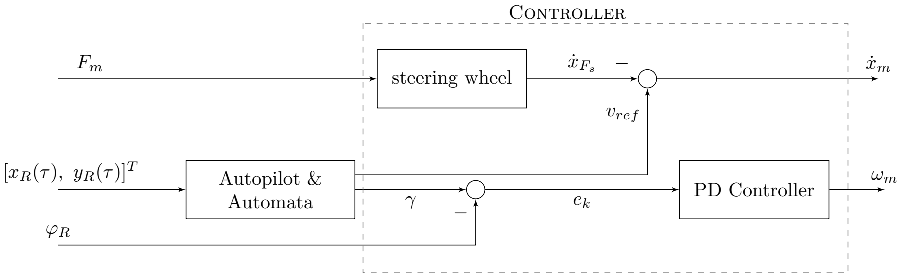


Figure 4-7: MATLAB block description

4-1-5 Algorithm-specific parameters

4-1-6 Parametric Risk Field

To take into account the data provided by the sensors, a gain-matrix $\mathbb{K}$ is defined as $\mathbb{K} = [0.06\ 0.2\ 0.5\ 1\ 1\ 0.5\ 0.2\ 0.06\ ]$ and $\alpha = 0.1$, this represents a collision avoidance seeking behavior.

Vector Field Histogram +

As suggested by the authors in [17], in order to select the steering direction a good set of parameters for a goal-oriented mobile robot is: $\mu_1 = 5$, $\mu_2 = 2$, and $\mu_3 = 2$.

4-1-7 Wave transformation

The virtual impedance parameter b was selected as $b = 10$.

4-1-8 Complete setup

The pose of the robot $\left([x_o,\ y_o,\ \varphi_R]^T\right)$ and the feedback force F_s are the inputs for the controller shown in Figure 4-7. A predefined goal location is used for the autopilot. Based on this data, the direction is determined using the algorithms discussed in Section 3-1. The controller works based on the orientation error, which is computed and then fed to the PD controller which converts this error into a desired angular velocity ω_m. The linear velocity of the robot, $\dot{x}_m$ is computed as the difference of the reference velocity v_{ref}, that comes from the Automata in Figure 3-1, and the velocity resulting from the feedback force $\dot{x}_{F_s}$, obtained by mapping the force F_s to $\dot{x}_s$ through a simulated steering wheel with a mass of 1 kg. The deadband compression algorithm is applied and two conditions are derived from it, i.e. whether or not the value must be sent. If no value is sent, the returning wave $v_m(t-\tau) = v_m(t_k - \tau)$, i.e. the previous received value. Once all the subsystems have been explained, the block diagram in Figure 4-8 shows the complete test setup.

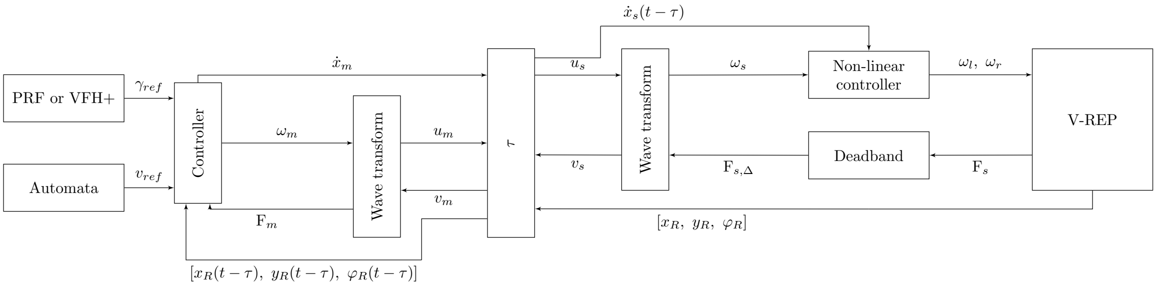


Figure 4-8: Teleoperation and collision avoidance implementation

4-2 Experiments

Data Collection

The data to be captured for each of the experiments is described in Table 4-5:

No.	Variable	Units	Characteristics
1.	Bandwidth	bits per second	Required communication capacity
2.	Standard deviation (σ)		Quality of data
3.	Variance (Var)		Quality of data
4.	Mean time	seconds	Simulation time
5.	Reduction	%	Reduction percentage
6.	Collisions	%	Percentage of simulations stopped by collisions

Table 4-5: Simulation data definition

The bandwidth value resembles the amount of data created per second. It is computed by storing the data that needs to be sent over the communication channel, i.e. the angular velocity ω_m. This still needs to be inserted in a packet and sent over the communication network; the size of this packets varies depending on the communication architecture and security, therefore only the data created by the algorithm is analyzed. The source figures for the dataset here capture are presented in Section A-1.

4-2-1 Simulation sets

In order to obtain datasets of required transmitted data and system's performance for comparison, the scheme shown in Figure 4-8 is used in different simulations.

In order to test the performance of the system under delay, a delay $\tau = 200$ ms is introduced as shown in Figure 4-8. To test the compression algorithms, the deadband approach is applied to the feedback force F_s, as shown in Figure 4-8. This feedback force is intended to be applied to a haptic controller, i.e. a steering wheel with force feedback. The characteristics of each of the simulations are explained below:

- **No delays - No compression** The first simulation set will be ran with $\tau = 0$, i.e. the ideal case with no delays on the communication channel and a relative deadband $\Delta_{u_{m,s}} = 0$, i.e. no compression. A set of 20 simulations are performed for each of the maps.
- **200 ms of delay - No compression** a delay $\tau = 200$ ms is introduced in order to test the performance of the system under such circumstances. The relative deadband value remains equal to zero.
- **200 ms of delay + Compression** The delay is kept at $\tau = 200$ ms and the compression is tested with two algorithms:
  - the relative deadband value $\Delta_{u_{m,s}} = 0.2 \times |u_{m,s}(t-1)|$.
  - the constant deadband value $\Delta_{u_{m,s}} = 0.1$, which accounts for a 10% difference in the force F_s, lining up with the data on Table 3-2.
- **200 ms of delay + Compression + measurement noise** Here, measurement noise d is introduced on the sensors as shown in Figure 4-9 to generate obstacle uncertainty and measure the system's performance. The reason to test the system's performance

under such uncertainties is to take into account some possible remote environmental characteristics that may affect the sensor's performance, e.g. UV rays, high vibrations, etc. The noise is composed of a normal distribution of random values in the following sets:

1. $|d| \leq 0.01$ m
2. $|d| \leq 0.05$ m
3. $|d| \leq 0.1$ m
4. $|d| \leq 0.25$ m
5. $|d| \leq 0.5$ m

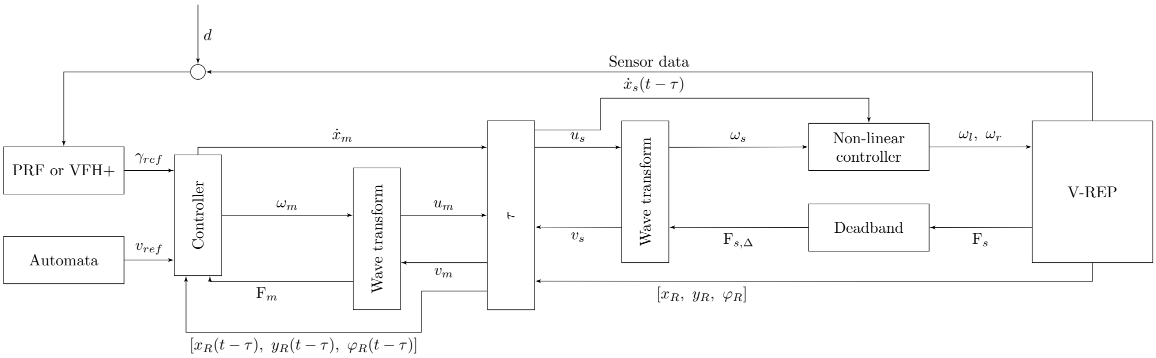


Figure 4-9: Measurement uncertainty on the sensors

Chapter 5

Results and Discussion

5-1 Results

5-1-1 Benchmark dataset

As discussed before, this simulation set is executed under ideal circumstances, i.e. without delays or compression algorithms. The results show that both algorithms create the same amount of data (a bandwidth of ≈ 630 bits per second (bps) is required) and therefore a comparison after the compression algorithms are applied will be based in this number. The complete results are shown in Table 5-1 to 5-3. It can be clearly noted that as the complexity of the map increases, so does the simulation times (going from 24 [s] up to 33 [s]). Furthermore, there is a noticeable increase in the time for the Vector Field Histogram + (VFH+) algorithm.

Variable	PRF	VFH+
Bandwidth	631 b/s	634 b/s
(σ)	3.76	2.01
Var	14.13	4.07
Time	23.67 s	23.73 s
Reduction	N/A	N/A
Collisions	0%	0%

Table 5-1: Simulation results for Map 1

Variable	PRF	VFH+
Bandwidth	621 b/s	633 b/s
(σ)	3.38	6.06
Var	11.45	36.81
Time	24 s	26.29 s
Reduction	N/A	N/A
Collisions	0%	0%

Table 5-2: Simulation results for Map 2

Variable	PRF	VFH+
Bandwidth	622 b/s	634 b/s
(σ)	3.72	1.4455
Var	13.39	2.09
Time	25.82 s	33.53 s
Reduction	N/A	N/A
Collisions	0%	0%

Table 5-3: Simulation results for Map 3

5-1-2 Performance with delay

Once a delay $\tau = 200$ ms in the communication channel has been introduced, the system's performance is measured again, this time the number of trials were reduced from 20 to 10 due to computation times. If a collision occurred during the testing, the simulation is stopped and registered as unsuccessful due to collision. The results collected for the simulation with delay for both algorithms behave similarly to the results without delay. This was expected due to the passivity properties of the wave transformation. The bandwidth remains bounded (in the neighborhood of 630 bps) and simulation times below 30 seconds. Here, the VFH+ algorithms continues to represent higher simulation time in comparison to the Parametric Risk Field (PRF) algorithm. These results are shown in Table 5-4 to 5-6.

Variable	PRF	VFH+
Bandwidth	630 b/s	632 b/s
(σ)	4.12	8
Var	16.98	64
Time	24.34 s	23.64 s
Collisions	0%	0%

Table 5-4: Simulation results for Map 1

Variable	PRF	VFH+
Bandwidth	624 b/s	635.85 b/s
(σ)	2.13	1.65
Var	4.55	2.2842
Time	25.55 s	33.85 s
Collisions	0%	0%

Table 5-6: Simulation results for Map 3

Variable	PRF	VFH+
Bandwidth	622 b/s	634.29 b/s
(σ)	4.66	5.22
Var	21.71	27.30
Time	24.5 s	27.58 s
Collisions	0%	0%

Table 5-5: Simulation results for Map 2

5-1-3 Delay + Deadband

Once the system's behavior under delay was assessed, two passivity-based compression methods are applied in order to verify their performance and the whole system's performance: a relative deadband of $\Delta_{u_{m,s}} = 0.2u_{m,s}(t-\tau)$ and a constant deadband $\Delta u_{m,s} = 10\% F_{m,s}$ are used on the feedback force F_S.

Relative Deadband

Following the conclusions of [34], the relative deadband shows compression rates of 95% – 97% for the PRF algorithm; the VFH+ paired up with the relative deadband has a compression rate of 42% – 53.5%. The results are shown in Table 5-7 to 5-9.

Table 5-7 to 5-9 are for the Relative Deadband while Table 5-10 to 5-12 show the data for the Constant deadband compression method.

Constant Deadband

A constant deadband of 10% of the feedback force is used following the just-noticeable difference (JND) in Table 3-2 as discovered by [36, 37]. The performance of the *constant deadband* compression method for both the PRF and the VFH+ algorithms remains similar to the previous algorithm. These results are shown in Table 5-10 to 5-12.

Variable	PRF	VFH+
Bandwidth	29.12 b/s	366.62 b/s
(σ)	1.94	11.10
Var	3.79	123.40
Time	23.94 s	24.96 s
Reduction	95.26%	41.94%
Collisions	0%	0%

Table 5-7: Simulation results for Map 1

Variable	PRF	VFH+
Bandwidth	29.12 b/s	388.85 b/s
(σ)	1.94	15.65
Var	3.79	245
Time	23.94 s	25.27 s
Reduction	95.26%	38.53%
Collisions	0%	0%

Table 5-10: Simulation results for Map 1

Variable	PRF	VFH+
Bandwidth	23.33 b/s	293.11 b/s
(σ)	9	14.37
Var	81	206.63
Time	24.5 s	27.58 s
Reduction	96.25%	53.79%
Collisions	0%	0%

Table 5-8: Simulation results for Map 2

Variable	PRF	VFH+
Bandwidth	23.33 b/s	348.3 b/s
(σ)	9	24.55
Var	81	603
Time	24.5 s	26.56 s
Reduction	96.25%	44.40%
Collisions	0%	0%

Table 5-11: Simulation results for Map 2

Variable	PRF	VFH+
Bandwidth	18.1 b/s	295.44 b/s
(σ)	2.82	7.27
Var	7.99	52.83
Time	25.55 s	33.85 s
Reduction	97.10%	53.54%
Collisions	0%	0%

Table 5-9: Simulation results for Map 3

Variable	PRF	VFH+
Bandwidth	18.1 b/s	303 b/s
(σ)	2.82	10.96
Var	7.99	120.3
Time	25.55 s	31.71 s
Reduction	97.10%	52.25%
Collisions	0%	0%

Table 5-12: Simulation results for Map 3

5-1-4 Performance under measurement noise

The results to be evaluated in this section are mainly the results obtained through the force feedback algorithm under high measurement noise circumstances, as explained before. The results are shown in Table 5-13 and 5-14 for the PRF algorithm and Table 5-15 and 5-16 for VFH+.

The robustness of both algorithms was tested here, and the performance decreased as the uncertainty increased as collisions appeared in the simulations. The collisions happened more often with objects in the environment and not with the surroundings, i.e. the walls. Both algorithms exhibited the same behavior under 0.5 [m] of uncertainty for the sensors: unavoidable collisions.

Uncertainty	$\mathbf{C}_R$	$\mathbf{R}_R$	$\mathbf{C}_C$	$\mathbf{R}_C$
0.01 m	0%	95%	0%	93.7%
0.05 m	10%	95%	0%	93.48%
0.1 m	0%	93.9%	0%	91.17%
0.25 m	100%	75%	100%	80.85%
0.5	100%	76%	100%	69.74%

Table 5-13: Results for Map 1 with PRF.

	Map 2				**Map 3**			
Uncertainty	$\mathbf{C}_R$	$\mathbf{R}_R$	$\mathbf{C}_C$	$\mathbf{R}_C$	$\mathbf{C}_R$	$\mathbf{R}_R$	$\mathbf{C}_C$	$\mathbf{R}_C$
0.01 m	0%	97.24%	0%	95.76%	0%	96.81%	0%	95.80%
0.05 m	0%	96.6%	0%	94.82%	0%	96.82%	10%	95.23%
0.1 m	20%	94.9%	0%	93.16%	70%	94.88%	70%	87.94%
0.25 m	100%	81%	100%	79.82%	100%	72%	100%	79%
0.5	100%	78%	100%	70.56%	100%	78.82%	100%	76%

Table 5-14: Results for Map 2 and Map 3 with PRF.

Uncertainty	$\mathbf{C}_R$	$\mathbf{R}_R$	$\mathbf{C}_C$	$\mathbf{R}_C$
0.01 m	0%	42%	0%	42.57%
0.05 m	0%	39.73%	0%	42.88%
0.1 m	0 %	42.58%	0%	46.93%
0.25 m	10%	53.6%	10%	62.7%
0.5	60%	60%	40%	73%

Table 5-15: Map 1 with VFH+.

	Map 2				Map 3			
Uncertainty	$\mathbf{C}_R$	$\mathbf{R}_R$	$\mathbf{C}_C$	$\mathbf{R}_C$	$\mathbf{C}_R$	$\mathbf{R}_R$	$\mathbf{C}_C$	$\mathbf{R}_C$
0.01 m	0%	49.70%	0%	50.63%	0%	50.73%	0%	44.68%
0.05 m	0%	44.15%	0%	47.05%	10%	47.70%	0%	42.07%
0.1 m	20%	40%	0%	42.40%	20%	41%	0%	41.72%
0.25 m	30%	47%	0%	48.88%	20%	43.27%	20%	49.81%
0.5	90%	56%	100%	56%	90%	56%	100%	58.5%

Table 5-16: Results for Map 2 and Map 3 with VFH+.

5-2 Discussion

5-2-1 Virtual forces

Throughout the experiments, the two different methods to create virtual forces, PRF and VFH+ have been assessed under close-to-real deployment circumstances. As the purpose of the virtual forces is to provide with an insight of the remote environment to the operator in order to avoid collisions, robustness of such an algorithm is appreciated.

From the previous experiments it can easily be observed that the compression rates are greater for the PRF algorithm in comparison to the VFH+. Upon examination of the forces created (see Figure 5-1 for a reference), it can be seen that the virtual force created by the PRF algorithm is smoother than the one created through the VFH+ algorithm. Therefore, greater compression rates are expected for this method.

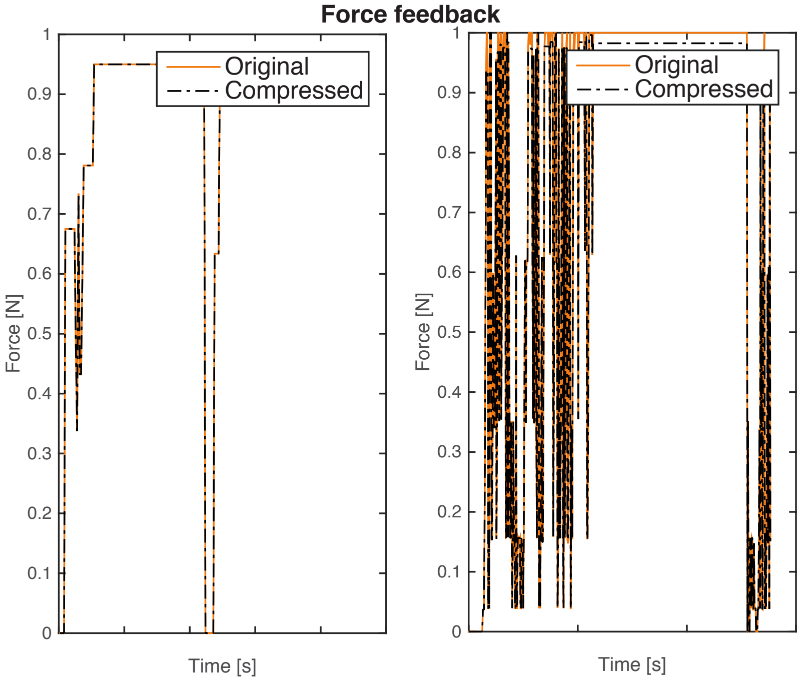


Figure 5-1: PRF (left) vs. VFH+ (right) virtual force

5-2-2 Compression methods

Benchmark set As seen in Table 5-1 to 5-3, the first experiment successfully creates a benchmark dataset. The information created in each of the three experiments is close enough to each other to work as a reference to compare the data savings when applying the compression methods. An average of 630 bits per second is created as a feedback force. The standard deviation for the bandwidth is $\sigma \leq 0.95$, which further indicates a trustworthy measure. Therefore, any reduction on the 630 bps benchmark is considered as an improvement in the following experiments.

Delay management While testing the system's performance over a delayed communication channel, the system's performance decrease is barely noticeable. Tables 5-4 to 5-6 show the stability properties derived from the domain switch to the wave-variables domain, as the data captured remains closely to the benchmark set in Section 5-1-1. Up to this point, no compression algorithms have been implemented. Simulation times vary slightly without making them highly relevant.

Compression The data presented in Table 5-7 to 5-9 show how an event-based control approach, i.e. the relative deadband approach, greatly decreases the amount of data created for the feedback force. The action the deadband takes upon the feedback force can be considered similar to a low-pass filter. As an example, in Figure 5-2 the comparison between the force feedback created at the robot's side vs. the force feedback applied to the steering wheel is shown; clearly resembles a low-pass filter.

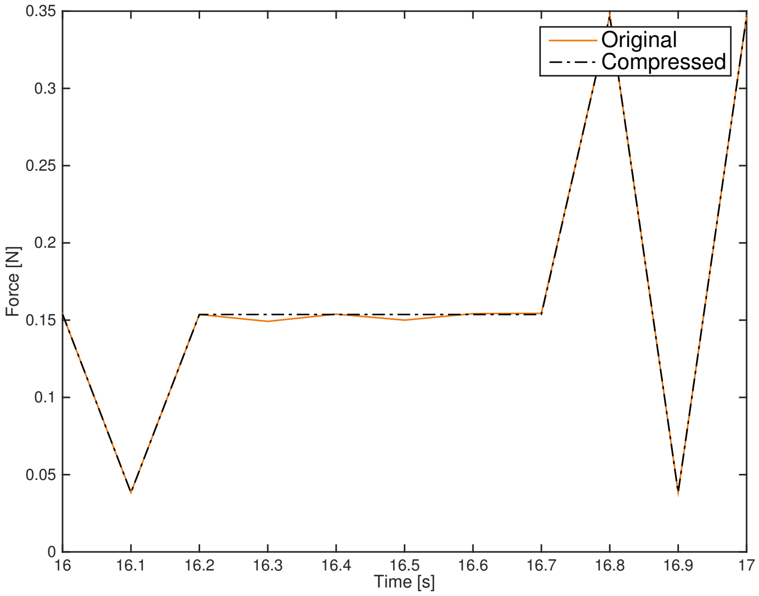


Figure 5-2: Deadband compression in action

From the tables, it is obvious how the compression achieves better results with the PRF algorithm when compared to the VFH+ algorithm, going from 95%+ of compression rates, in words this means that 95% of the original data is discarded. These results were achieved under ideal circumstances, i.e. no measurement noise d. To further test the system's performance under closer-to-real circumstances, a disturbance is added and evaluated next.

5-2-3 Robustness

The results for the system's performance under high measurement uncertainty as described in Section 4-2-1. Here the collision and compression rates are the basis for the evaluation.

Following the performance in the previous simulations with ideal measurement conditions, the compression is highest with the PRF algorithm. However, this algorithm does not perform well under high measurement uncertainties, $d \geq 0.1$m: the algorithm cannot distinguish properly between measurement noise and obstacles in the environment and therefore shows a less-smoother trajectory which ends up in a collision for $d \geq 0.25$ [m]. It is also observed that the compression rates decrease greatly under high uncertainty as the force's fluctuation frequency increases dramatically as a consequence of the noise.

The VFH+ algorithm is able to handle the uncertainty and avoid collisions better than the PRF and up to an uncertainty of $d \geq 0.5$m. Data compression rates remain constant for the VFH+ algorithm (ranges between 40% to 50% of compression rates).

While there is a performance deterioration in comparison to the previous experiments under ideal circumstances, the VFH+ certainty-based grids clearly outperforms the PRF under high measurement uncertainty, based on the collision rate.

Chapter 6

Conclusions and Recommendations

6-1 Conclusions

Based on the results of the previous chapter, the following comparison between the collision avoidance algorithms can be derived:

- The Parametric Risk Field (PRF) algorithm generates a smoother feedback force, which based on the compression characteristics of the event-based control algorithm, i.e. dead-band, a smoother function allows for greater compression.
- Both methods are able to create scalable forces, therefore adjustments can be performed in order to suit specific haptic feedback controllers and fitted to operator's preferences.
- The domain switch from the Mechanical domain (forces and velocities) to the wave-domain (u and v) based in the passivity-preservation scheme, is less influenced by distortions created by the delays in the communication channel and therefore, is able to perform under delay up to 200 [ms].
- Under low uncertainty ($d \leq 0.1$ [m]) the PRF algorithm is best suited as it achieves better compression rates, and a smoother feedback force generation. If high uncertainty (measurement noise) is present, the PRF is no longer the best suitable option, as it cannot eliminate the effect of the disturbance d properly, and therefore the Vector Field Histogram + (VFH+) algorithm becomes a viable option.
- There is a trade-off on the performance in high uncertainty (VFH+ algorithm) and the compression rates (achieved with the PRF algorithm.

The implementation of the VFH+ algorithm as a force feedback generator yielded promising results. The compression rates can be increased through passive compression methods as the ones presented in Section B-1-2. The next step for this algorithm is to be implemented in a real setup and measure the performance with an operator, specifically workload and ease of use, in order to line up with the goals of this project.

6-2 Recommendations

Force magnitude In the experiments used for this project, the force magnitude was selected from 0 to 1 for as an example and to lower the computational requirements, however in a physical implementation of the methods here discussed, a proper scalability for the force must be performed prior its use. The output of the feedback force algorithms here discussed can easily fit any force magnitude and resolution of the haptic controller.

Because the feedback force generated by the two algorithms here compared differ in the smoothness, it is interesting to compare the workload on the human operator due to this feedback force.

Bandwidth requirements As discussed before, the results here analyzed only take into account the information generated by the algorithm, without taking into account TCP/IP or similar communication protocols for the communication, as they depend on the properties of specific communication methods and security protocols. The data savings here exposed follow a scenario on which the communication protocols are already implemented for other data (e.g. audio and/or video streams) and haptic data is attached to them, representing savings on the haptic data communication.

Deadband extension to multiple Degrees of Freedoms (DoFs) The scope of this project was to test these algorithms on robots that are part of the Special Euclidean group $SE(2)$, i.e. with three DoF. This setup required a 1 DoF deadband implementation. Research has been done regarding the extension of these compression algorithms to multiple DoFs, e.g. the multi-dimensional approach proposed in [6], based on the validity of a straightforward extension of Weber's Law to multiple dimensions. In two dimensions, the deadband then becomes a circular area. In 3-D, a spherical volume element serves as the deadzone p (multiple-dimension deadband). The new criterion for transmission is based in vectors $\mathbf{v}_i$ and $\mathbf{v}_c$, denoted in bold letters; d represents the magnitude of their difference. Accordingly, the deadzone criteria reads as follows:

$$\begin{aligned} & d = |\mathbf{v}_i - \mathbf{v}_c| \\ & d \leq p \cdot |\mathbf{v}_i| \rightarrow \text{Do nothing} \\ & d > p \cdot |\mathbf{v}_i| \rightarrow \text{Transmit new vector } \mathbf{v}_c \end{aligned} \tag{6-1}$$

An extended explanation can be found in Section B-1.

Delays The domain transformation resulted in a system that is able to perform normally under delays up to 200 [ms]. However, it would be interesting to study the behavior of the system with higher delays. The virtual impedance b used in the wave transformation could be increased, resulting in a faster system that could perform under higher communication delays, however the system's stability should be studied and the energy function monitored.

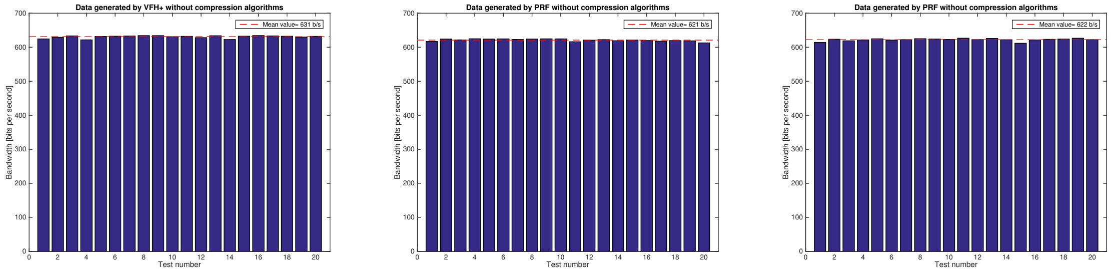
Appendix A

Figures

A-1 Experimental Figures

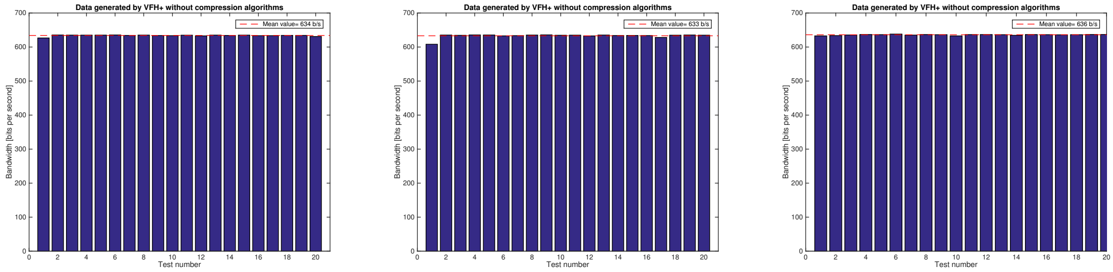
A-1-1 Benchmark dataset

The following figures show the bandwidth used by each one of the simulations with ideal circumstances, i.e. without noise nor delays on the communication channel. Figure A-1 shows the bandwidth for the Parametric Risk Field (PRF) algorithm while Figure A-2 shows the Vector Field Histogram + (VFH+) algorithm.



(a) Map 1 **(b)** Map 2 **(c)** Map 3

Figure A-1: Bandwidth required for the PRF algorithm.



(a) Map 1

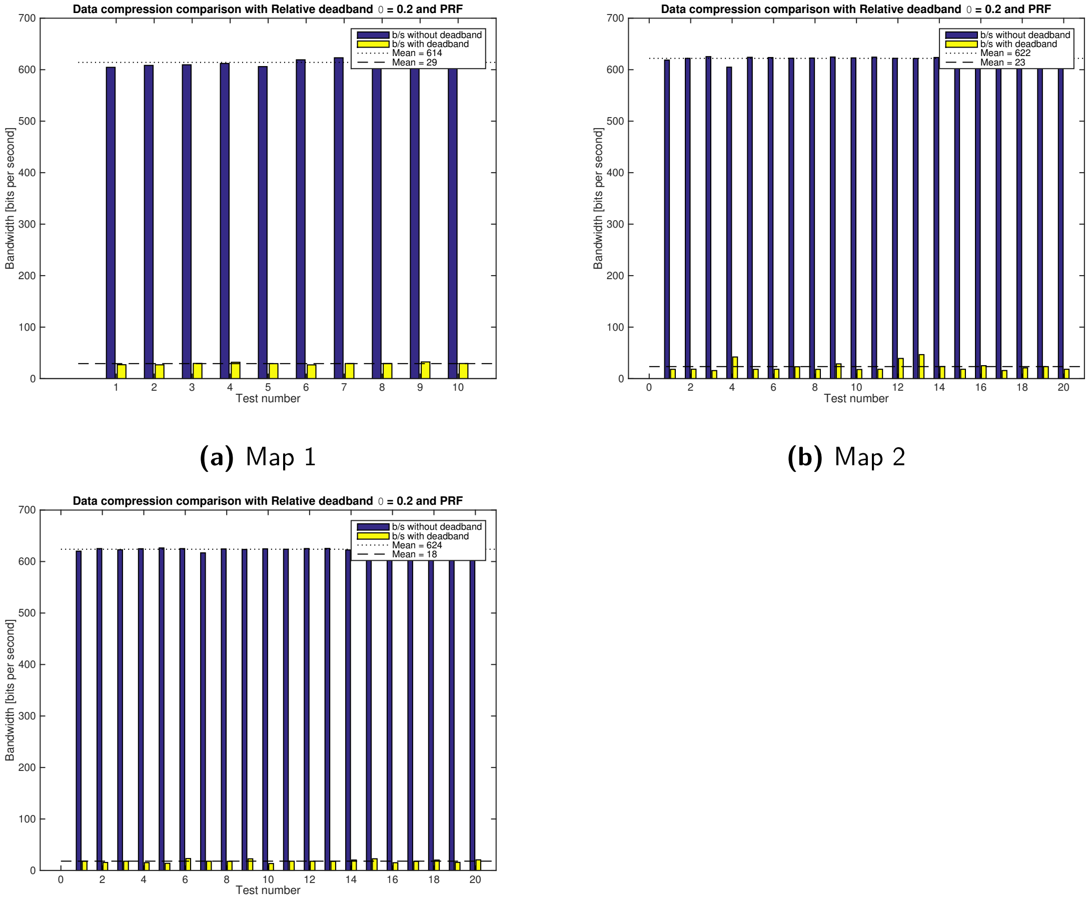
(b) Map 2

(c) Map 3

Figure A-2: Bandwidth required for the VFH+ algorithm.

A-2 Compression rates

In the following Figures, comparisons between the uncompressed and compressed data are shown. Lining up with the results achieved in Chapter 5.

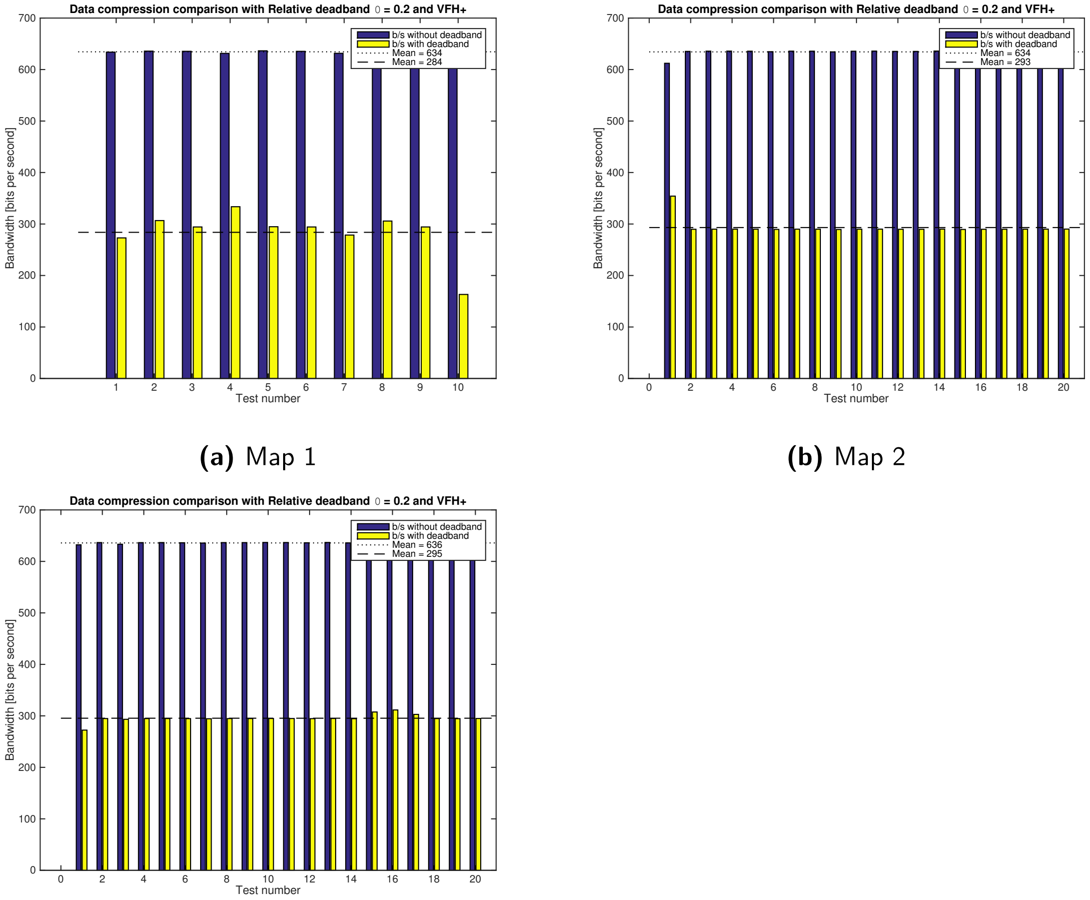


(a) Map 1

(b) Map 2

(c) Map 3

Figure A-3: Compression achieved (in yellow) for the PRF algorithm with delay $\tau = 200$ [ms].



(a) Map 1

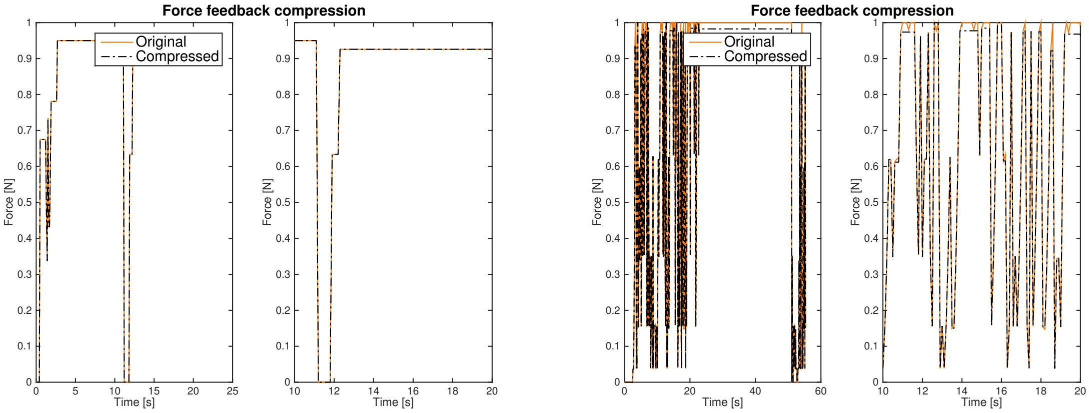
(b) Map 2

(c) Map 3

Figure A-4: Compression achieved (in yellow) for the VFH+ algorithm with delay $\tau = 200$ [ms].

A-3 Force feedback

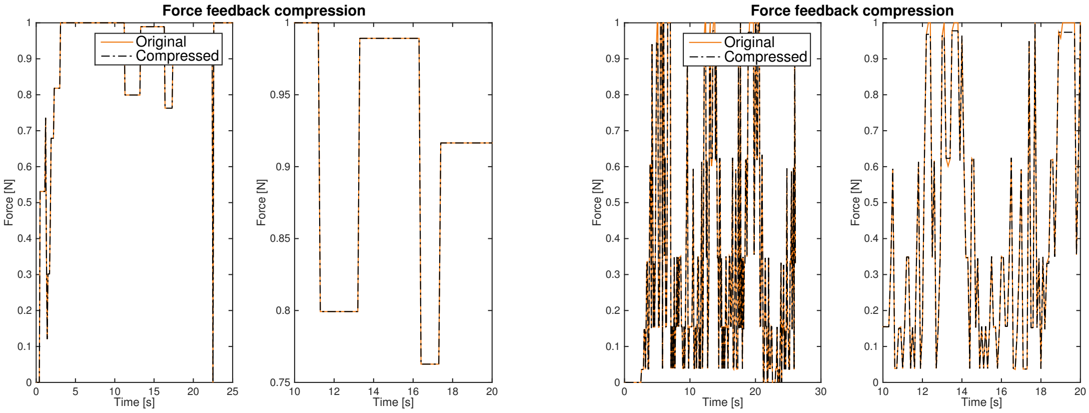
In this section, examples of the generated feedback force F_s under different circumstances are shown.



(a) Map 1

(b) Map 2

Figure A-5: Feedback force and compression for the PRF (left) and VFH+ (right) algorithms with delay $\tau = 200$ [ms].



(a) Map 1

(b) Map 2

Figure A-6: Feedback force and compression for the PRF (left) and VFH+ (right) algorithms with delay $\tau = 200$ [ms] and uncertainty $d \leq 0.10$ [m].

Appendix B

Elaboration in future work

In this section, an elaboration on the tools used throughout the project is included. These tools were not used in the current project as they fall out of the scope of it, however they are relevant for future work on this project as they would provide tools to extend the project into other tasks such as dexterous tasks.

B-1 Deadband

B-1-1 Deadband extension to multiple Degrees of Freedom

To overcome this limitation, a multi-dimensional approach is proposed in [6], based on the validity of a straight-forward extension of Weber's Law to multiple dimensions. In two dimensions, the deadband then becomes a circular area as shown in Figure B-1. In 3-D, a spherical volume element serves as the deadzone p (multiple-dimension deadband). The new criterion for transmission is based in vectors $\mathbf{v}_i$ and $\mathbf{v}_c$, denoted in bold letters; d represents the magnitude of their difference. Accordingly, the deadzone criteria reads as follows:

$$\begin{aligned} &d = |\mathbf{v}_i - \mathbf{v}_c| \\ &d \leq p \cdot |\mathbf{v}_i| \rightarrow \text{Do nothing} \\ &d > p \cdot |\mathbf{v}_i| \rightarrow \text{Transmit new vector } \mathbf{v}_c \end{aligned} \tag{B-1}$$

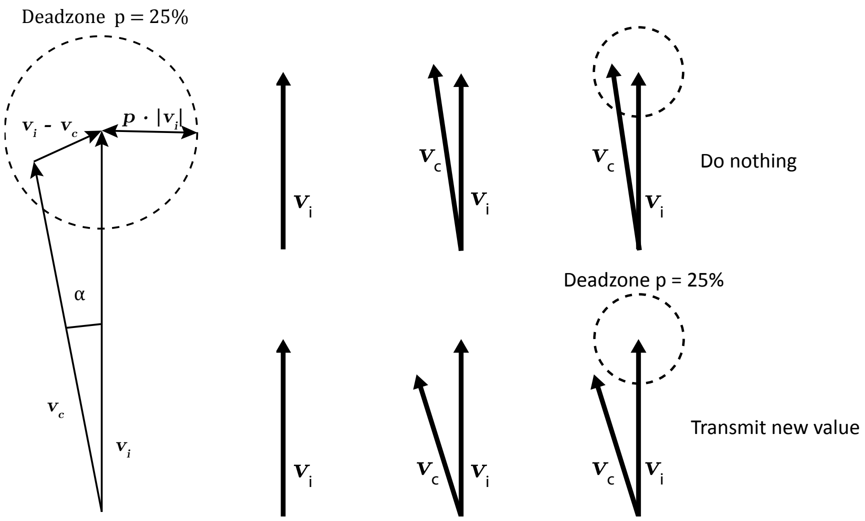


Figure B-1: Two-dimensional deadband (deadzone)

To compute the deadzone, the size of the circle depends on the length of the vector $\mathbf{v}_i$ and the maximum of the angle α depends on the deadzone factor p. The angle α then reaches its maximum when $\mathbf{v}_c$ is tangential to the deadzone circle, as shown in Eq. (B-2)

$$\begin{aligned} &\mathbf{v}_c \perp \mathbf{v}_i - \mathbf{v}_c \\ &\text{and} \\ &|\mathbf{v}_i - \mathbf{v}_c| = p \cdot |\mathbf{v}_i| \end{aligned} \tag{B-2}$$

If $p << 1$ is assumed, α_{max} can be computed as shown in Eq. (B-3):

$$\begin{aligned} \sin \alpha_{max} &= \frac{|\mathbf{v}_i - \mathbf{v}_c|}{|\mathbf{v}_i|} = \frac{p \cdot |\mathbf{v}_i|}{|\mathbf{v}_i|} = p \\ \alpha_{max} &= \arcsin\, p, \end{aligned} \tag{B-3}$$

and therefore $|\mathbf{v}_{c_{\alpha_{max}}}|$ would be:

$$\begin{aligned} \cos \alpha_{max} =& \frac{|\mathbf{v}_{c_{\alpha_{max}}}|}{|\mathbf{v}_i|} \\ |\mathbf{v}_{c_{\alpha_{max}}}| =& \cos \alpha_{max} \cdot |\mathbf{v}_i| \\ =& \cos(\arcsin\, p) \cdot |\mathbf{v}_i| \end{aligned} \tag{B-4}$$

Eq. (B-4) shows that the magnitude of a sample two-Degrees of Freedom (DoF) variable is not relevant, once it changes its direction by α_{max}, an update will be sent. Hence this algorithm is magnitude-independent, an important property of the isotropic deadzone applicable to this project, as the two DoFs considered here involve different magnitudes, m/s for v and rad/s for ω.

B-1-2 Passive compression methods

Model-Based prediction

To further on reduce the transmitted packets, a linear signal prediction is used in both sides of the teleoperation system as shown in Figure B-2. On the operator's side, a force predictor is used to estimate the future force values based on interpolation, on the teleoperator (robot's) side, the predictor uses the sent values, an extrapolative method. These methods are further on discussed in Section B-1-2. As an example, the authors of [6] propose the following linear predictor (B-5):

$$v_i = \begin{cases} v_{\text{new}} & \text{value sent/arrived} \\ v_{i-1} + \frac{v_{\text{new}-1} - v_{\text{new}-2}}{t_{\text{new}-1} - t_{\text{new}-2}} \cdot (t_i - t_{i-1}) & \text{else} \end{cases} \tag{B-5}$$

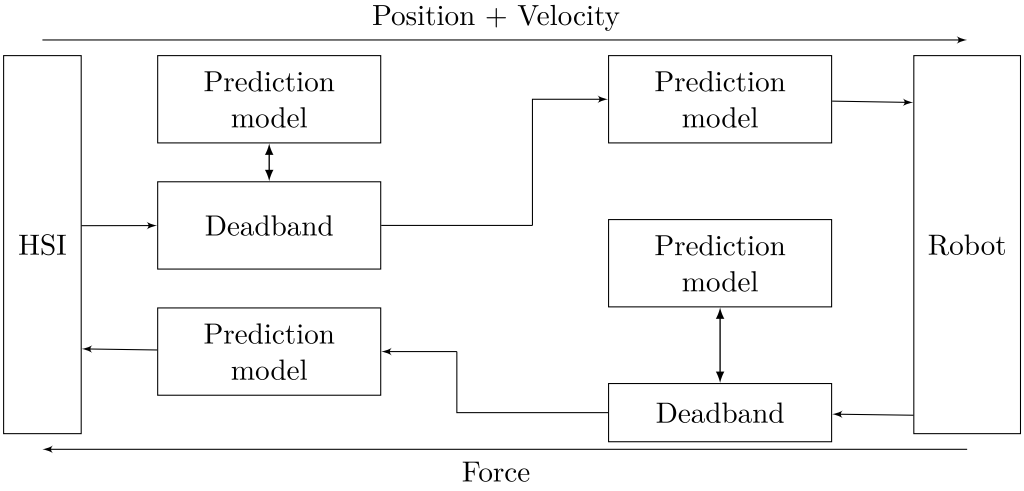


Figure B-2: System with deadzone and model-based prediction

Lossy compression methods

Passive interpolative downsampling Kuschel et al. [54], present two lossy-data-compression (LDC) algorithms, these are a frame-based passive interpolative downsampling (iDS) and a sample-based passive extrapolative downsampling (eDS). In order to apply these algorithms, the authors make use of scattering techniques described below in Section 2-6. These methods make used of an energy-supervision function to preserve stability of the system.

Interpolative Compression Stategy The main idea behind this method, is to approximate the original signal by its mean value, i.e. a number of k_F samples are accumulated to a frame, then the parameter vector $\mathbf{p}$ is replaced by a single element, which corresponds to its mean value (Eq. (B-6)) within the interpolation frame t_F and the resulting value is transmitted over the network.

$$\mathbf{p} = \frac{1}{T_F}\int_{t_j}^{t_j+T_F} u_r(t) \quad \forall t_F = [t_j, t_j+T_F] \tag{B-6}$$

where T_F is an induced delay and it is defined as:

$$T_F = \frac{k_F}{f_s} \tag{B-7}$$

where f_s is the sampling frequency and t_j the starting time of the frame j. It becomes obvious then, that the compression rate of communications is defined as $CR_{iDS} = k_F : 1$. This resembles a low-pass filter. Interpolating with the mean value yields minimal energy if the waves remain constant, making the passivity condition in Eq. (2-41) true, and therefore guaranteeing stability of the system. Therefore, a condition that should be monitored for the iDS (an extension of Eq. (2-41)) is shown in Eq. (B-8):

$$\int_{t_j}^{t_j+T_F} u_s^2(t)dt \geq \int_{t_j+T_F+\tau}^{t_j+2T_F+\tau} u_m^2(t)dt \tag{B-8}$$

The compression method relies on the size of the vector k_F: the longer it is, the higher the compression rate CR, however, as shown in Figure B-3, there exists an induced delay T_F (B-7) and a reduction on the system's transparency; there is a trade-off between the compression and the transparency.

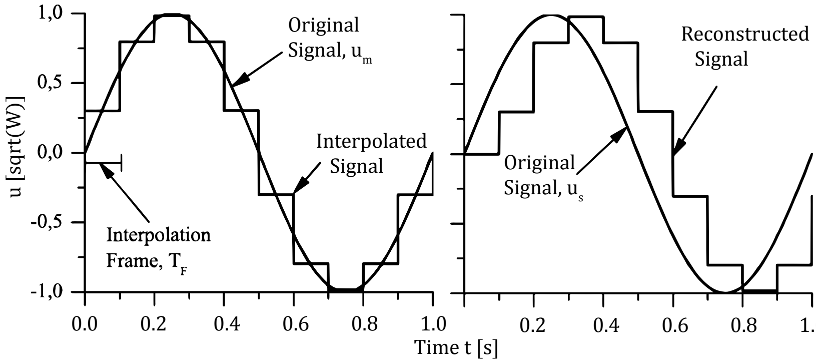


Figure B-3: Principle of interpolative compression iDS.

The advantages of this method in comparison to the Deadband method shown in Section 3-2-2 are:

- Freely adjustable data rate, i.e. the size of k_F and the sampling frequency f_s can be adjusted to satisfy communication and bandwidth limits,
- arbitrary algorithms, apart from the mean value, can be used as long as the energy output condition in Eq. (B-8) is satisfied.

Extrapolative Compression Strategy This strategy estimates future samples up to an estimation horizon T_{EH}; this method resembles Model Predictive based controllers, without the need of a model. It works as follows: a number of k_{EH} are estimated and an encoded signal is constructed based on assumptions, resulting in a parameter vector $\mathbf{p}$ which is sent over the network and reconstructed by the decoder. For every k_{EH} samples, the estimation for the following k_{EH} values is computed. The extrapolation horizon is computed in a similar way as the induced delay in the iDS method:

$$T_{EH} = \frac{k_{EH}}{f_s} \tag{B-9}$$

The compression principle is shown in Figure B-4.

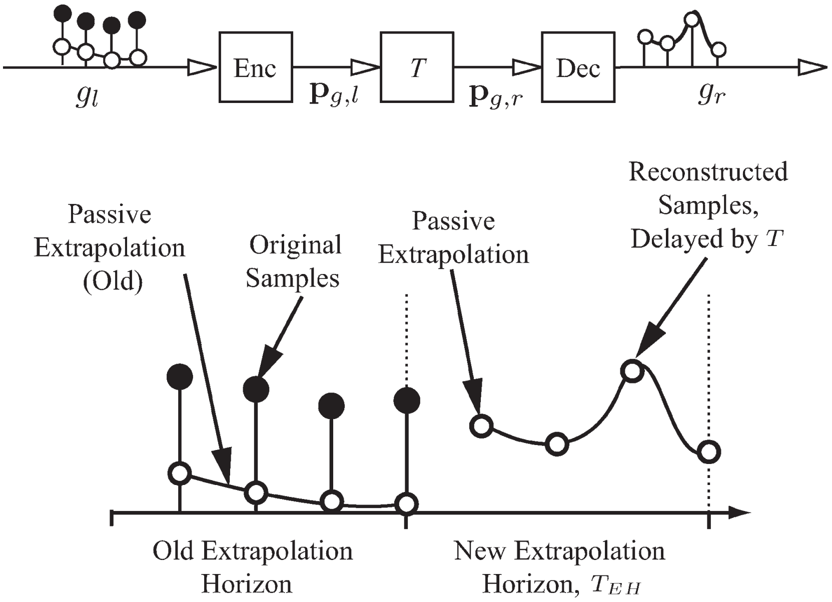

g_l
Enc
$\mathbf{p}_{g,l}$
T
$\mathbf{p}_{g,r}$
Dec
g_r
Passive Extrapolation (Old)
Original Samples
Passive Extrapolation
Reconstructed Samples, Delayed by T
Old Extrapolation Horizon
New Extrapolation Horizon, T_{EH}

Figure B-4: Principle of extrapolative compression strategy: eDS.

Similar to the iDS method, the passivity condition derived from Eq. (2-41) is shown in Eq. (B-10).

$$\int_0^{t_j} u_r^2 dt - \int_0^{t_j+T} u_l^2 dt \geq \int_{t_j+T}^{t_j+T_{EH}+T} u_l^2 dt \tag{B-10}$$

The main idea of this algorithm is to extrapolate the future signal by a single value. There are two possibilities: if the last value measured satisfies the passivity criterion (B-10), then it is considered as extrapolation through the extrapolation horizon. If on the other hand, the passivity criterion does not hold, i.e. energy was created, then a value reduced in energy is computed such that (B-10) holds. Therefore the active passivation is defined as:

$$\mathbf{p}_{g,l} = \begin{cases} u_r(t-1) & \text{if } (B-10) \text{ holds} \\ u_r & \text{such that (B-10) holds} \end{cases} \quad \forall t = [t_j,\ t_j + T_{EH}] \tag{B-11}$$

B-2 Traps and Virtual Force Fields

Virtual Force Field (VFF) algorithms such as the Parametric Risk Field (PRF) have one drawback: potential traps. Research on these kind of traps [12, 55] has been performed and throughout the research the following traps were derived:

1. **Trap situations due to local minima**. This trap may occur when the robot runs into a dead end.
2. **No passage between closely spaced obstacles**. When the robot attempts to go through to closely positioned obstacles, the repulsive forces generated by them cause an oscillation in the robot, rendering it difficult to control.
3. **Oscillations in the presence of obstacles**.
4. **Oscillations in narrow passages**.

Because of the shared-control approach in this project, these traps are negligible, however looking forward to a supervisory-control scheme on which the human operator will only interact with the robot when required, these traps must be taken into account. Below, an introduction to the potential solutions for these traps are presented.

B-2-1 Virtual Force Fields with Traps

In order to research further on methods that will provide a solution to the aforementioned traps, [55] take a deep look into mobile robot path planning and its application. To achieve this, the relative distance between the robot and the goal is also taken into account. This problem is then denoted as Goal Non Reachable with Obstacle Nearby (GNRON). This problem is due to the fact that as the robot approaches the goal, the repulsive potential increases as well in the presence of obstacles. To counteract this, a new repulsive potential function is constructed, taking into account the relative distance to the goal. Even though the proposed potential function assures that the goal is the global minimum of the total potential, it cannot eliminate other local minima due to obstacles between the robot and the goal.

B-2-2 Virtual Force Fields with Traps: Wall Follow Method

study the various VFF traps for automated navigation and finally present a plausible solution to the trap problems, this is by an trap-detection algorithm. This algorithm works by simply monitoring the speed of the robot. If the robot is caught in a trap, the speed will become zero. In a dynamic system, the robot will either oscillate or run in a closed loop. Therefore, the authors' proposition is to monitor the robot-to-target direction. If this direction is greater than 90°, the robot is moving away from the target and is likely to get trapped. Once this condition is detected, the robot switches to a recovery algorithm, namely the Wall-Following Method (WFM). This will allow the robot to follow the contour of the obstacle until it starts heading back into a direction less than 90° off-target.

Throughout the experiments, this new algorithm presented promising results until more than one trap was detected. In given case, the authors came with another method. Two possible WFMs can be used: L- or R-mode, i.e. when the robot follows a wall to its left or to its right. In a given run, the robot can select either method for the first trap situation, but then it must stick to the selected method for the whole run.

One last exception addressed by the authors is when the robot is on a WFM on an inside wall of a closed room. In this case the robot will follow that wall indefinitely since the condition to leave the WFM will never be reached. To detect this situation, the sum of the changes of the target direction (angle) is monitored. Therefore, when this summation is greater than 360° indicates that the robot has traveled at least one full loop around the target (and is currently on a trap). Once this is detected, the robot is forced out of the WFM and back into the normal target pursuit.

Bibliography

[1] European Space Agency, “1 dof setup,” 2015.

[2] I. Surgical, “The da vinci surgical system,” *Intuitive Surgical Inc., Sunnyvale, CA, available at: http://www. intuitivesurgical. com*, 2013.

[3] M. A. Lerner, M. Ayalew, W. J. Peine, and C. P. Sundaram, “Does training on a virtual reality robotic simulator improve performance on the da vinci® surgical system?,” *Journal of Endourology*, vol. 24, no. 3, pp. 467–472, 2010.

[4] D. Morris, H. Tan, F. Barbagli, T. Chang, and K. Salisbury, “Haptic feedback enhances force skill learning,” in *EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint*, pp. 21–26, IEEE, 2007.

[5] S. Shamsunder and M. Manivannan, “Haptic guided laparoscopy simulation improves learning curve,” *Studies in health technology and informatics*, vol. 132, p. 454, 2008.

[6] P. Hinterseer, S. Hirche, S. Chaudhuri, E. Steinbach, and M. Buss, “Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems,” *Signal Processing, IEEE Transactions on*, vol. 56, no. 2, pp. 588–597, 2008.

[7] S. Hirche, P. Hinterseer, E. Steinbach, and M. Buss, “Towards deadband control in networked teleoperation systems,” in *Proceedings IFAC World Congress, International Federation of Automatic Control*, Citeseer, 2005.

[8] E. Vander Poorten, E. Demeester, E. Reekmans, J. Philips, A. Hüntemann, and J. De Schutter, “Haptic obstacle avoidance for intuitive powered wheelchair navigation,”

[9] T. M. Lam, H. W. Boschloo, M. Mulder, and M. M. van Paassen, “Artificial force field for haptic feedback in uav teleoperation,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 39, no. 6, pp. 1316–1330, 2009.

[10] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[11] D. Xiao and R. Hubbold, "Navigation guided by artificial force fields," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 179–186, ACM Press/Addison-Wesley Publishing Co., 1998.

[12] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, no. 5, pp. 1179–1187, 1989.

[13] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 501–518, 1992.

[14] D. H. Kim and S. Shin, "Local path planning using a new artificial potential function composition and its analytical design guidelines," *Advanced Robotics*, vol. 20, no. 1, pp. 115–135, 2006.

[15] M. Mutlu, "Potential field navigation," 2012.

[16] H. W. Boschloo, T. M. Lam, M. Mulder, and M. Van Paassen, "Collision avoidance for a remotely-operated helicopter using haptic feedback," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 1, pp. 229–235, IEEE, 2004.

[17] I. Ulrich and J. Borenstein, "Vfh+: Reliable obstacle avoidance for fast mobile robots," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2, pp. 1572–1577, IEEE, 1998.

[18] S. M. Udupa, *Collision detection and avoidance in computer controlled manipulators.* PhD thesis, Citeseer, 1977.

[19] O. Ringdahl, "Path tracking and obstacle avoidance for forest machines," master's thesis, Department of Computing Science, Umeå University, April 2003. UMNAD 454/03.

[20] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 278–288, 1991.

[21] E. Garcia and P. J. Antsaklis, "Model-based event-triggered control with time-varying network delays," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pp. 1650–1655, IEEE, 2011.

[22] D. Yue, Q.-L. Han, and J. Lam, "Network-based robust h? control of systems with uncertainty," *Automatica*, vol. 41, no. 6, pp. 999–1007, 2005.

[23] C. Santos, M. Mazo Jr, and F. Espinosa, "Adaptive self-triggered control of a remotely operated robot," in *Advances in Autonomous Robotics*, pp. 61–72, Springer, 2012.

[24] C. Santos, M. Mazo, E. Santiso, F. Espinosa, and M. Martínez, "Adaptive self-triggered control for remote operation of wifi linked robots," in *ROBOT2013: First Iberian Robotics Conference*, pp. 541–554, Springer, 2014.

[25] D. Lehmann and J. Lunze, "Event-based control with communication delays," in *Proceedings of the 18th IFAC world congress*, 2011.

[26] K. J. Aström, “Event based control,” in *Analysis and design of nonlinear control systems*, pp. 127–147, Springer, 2008.

[27] D. Jeltsema, “Modeling & nonlinear systems theory,” publication, TU Delft, 2013.

[28] A. van der Schaft and D. Jeltsema, *Port-Hamiltonian Systems Theory: An Introductory Overview*. Now Publishers Incorporated, 2014.

[29] A. Van der Schaft, *L2-gain and passivity techniques in nonlinear control*. Springer Science & Business Media, 2012.

[30] G. Niemeyer and J.-J. Slotine, “Stable adaptive teleoperation,” *Oceanic Engineering, IEEE Journal of*, vol. 16, no. 1, pp. 152–162, 1991.

[31] M. C. Dede, S. Tosunoglu, and D. W. Repperger, “Effects of time delay on force-feedback teleoperation systems,” in *IEEE/CSS 12th Mediterranean Conference on Control Automation, IEEE/CSS MED?, Kusadasi, June*, pp. 6–9, 2004.

[32] G. T. R. Corporation and J.-P. de la Croix, “Sim.i.am.” https://github.com/jdelacroix/simiam, 2014.

[33] E. H. Weber, *Die Lehre vom Tastsinne und Gemeingefühle auf Versuche gegründet*. Friedrich Vieweg und Sohn, 1851.

[34] S. Hirche, P. Hinterseer, E. Steinbach, and M. Buss, “Network traffic reduction in haptic telepresence systems by deadband control,” in *Proceedings IFAC World Congress, International Federation of Automatic Control*, 2005.

[35] S. Hirche and M. Buss, “Human-oriented control for haptic teleoperation,” *Proceedings of the IEEE*, vol. 100, no. 3, pp. 623–647, 2012.

[36] C. A. Lawn and B. Hannaford, “Performance testing of passive communication and control in teleoperation with time delay,” in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pp. 776–783, IEEE, 1993.

[37] H. Z. Tan, N. I. Durlach, Y. Shao, and M. Wei, “Manual resolution of compliance when work and force cues are minimized,” *ASME DYN SYST CONTROL DIV PUBL DSC, ASME, NEW YORK, NY,(USA), 1993,*, vol. 49, pp. 99–104, 1993.

[38] M. F. E. Rohmer, S. P. N. Singh, “V-rep: a versatile and scalable robot simulation framework,” in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[39] ADEPT MOBILEROBOTS LLC, *Pioneer P3-DX*, 2015.

[40] J. Storms, S. Vozar, and D. Tilbury, “Predicting human performance during teleoperation,” in *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pp. 298–299, ACM, 2014.

[41] C. C. Macadam, “Understanding and modeling the human driver,” *Vehicle System Dynamics*, vol. 40, no. 1-3, pp. 101–134, 2003.

[42] D. Weir and A. Phatak, “Model of human response to step transitions in controlled element dynamics,” tech. rep., Technical Report NASA CR-671, 1968.

[43] R. Woodworth and H. Schlosberg, “Experimental psychology, rev. ed. new york: Holt, rinehart and winston,” 1954.

[44] W. H. Teichner, “Recent studies of simple reaction time.,” *Psychological Bulletin*, vol. 51, no. 2, p. 128, 1954.

[45] L. B. Palma, F. V. Coito, and P. S. Gil, “Low order models for human controller-mouse interface,” in *Intelligent Engineering Systems (INES), 2012 IEEE 16th International Conference on*, pp. 515–520, IEEE, 2012.

[46] M. Arif and H. Inooka, “Iterative manual control model of human operator,” *Biological cybernetics*, vol. 81, no. 5-6, pp. 445–455, 1999.

[47] J. G. Ziegler and N. B. Nichols, “Optimum settings for automatic controllers,” 1942.

[48] C. C. Hang, K. J. Åström, and W. K. Ho, “Refinements of the ziegler–nichols tuning formula,” in *IEE Proceedings D (Control Theory and Applications)*, vol. 138, pp. 111–118, IET, 1991.

[49] M. Schlegel, “Exact revision of the ziegler-nichols frequency response method,” in *Control and Applications*, ACTA Press, 2002.

[50] I. Boiko, *Non-parametric Tuning of PID Controllers: A Modified Relay-Feedback-Test Approach*. Advances in Industrial Control, Springer London, 2012.

[51] B. R. Fajen and W. H. Warren, “Behavioral dynamics of steering, obstable avoidance, and route selection.,” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 29, no. 2, p. 343, 2003.

[52] B. R. Fajen, W. H. Warren, S. Temizer, and L. P. Kaelbling, “A dynamical model of visually-guided steering, obstacle avoidance, and route selection,” *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 13–34, 2003.

[53] T. M. Lam, M. Mulder, and M. Van Paassen, “Collision avoidance in uav tele-operation with time delay,” in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pp. 997–1002, IEEE, 2007.

[54] M. Kuschel, P. Kremer, and M. Buss, “Passive haptic data-compression methods with perceptual coding for bilateral presence systems,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 39, no. 6, pp. 1142–1151, 2009.

[55] S. S. Ge and Y. J. Cui, “New potential functions for mobile robot path planning,” *IEEE Transactions on robotics and automation*, vol. 16, no. 5, pp. 615–620, 2000.

Glossary

List of Acronyms

UAV	Unmanned Aerial Vehicle
AFF	Artificial Force Field
GPF	Generalized Potential Field
PRF	Parametric Risk Field
VFF	Virtual Force Field
GNRON	Goal Non Reachable with Obstacle Nearby
WFM	Wall-Following Method
OG	Oil and Gas
HLS	Hold Last Sample
LDC	lossy-data-compression
iDS	passive interpolative downsampling
eDS	passive extrapolative downsampling
HSSE	Health, Safety, Security and Environment
VFH	Vector Field Histogram
VFH+	Vector Field Histogram +
RCP	Robot Center Point
DoF	Degrees of Freedom
HSI	Human-System Interface
ZOH	Zero Order Hold

IH	Impulse Hold
JND	just-noticeable difference
bps	bits per second