UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCATRONICA

TESI DI LAUREA MAGISTRALE

Online Recognition of Environment Properties by Using Bilateral Control

Relatore: Roberto Oboe

Laureando: Giacomo Pradella
1151573-IMC

ANNO ACCADEMICO: 2019-20

ABSTRACT

In this thesis the problem of identification of the mechanical impedance of an unknown environment is addressed. The topic is of great interest to enhance the performances in the control of robotic systems in an unstructured environment.

For performing tasks such as telemanipulation or grasping of delicate objects, the knowledge of the mechanical properties of the environment is required to ensure the integrity of the manipulated object.

Through the use of bilateral control based on DOB and RFOB structures, position, speed and force information are gathered and analyzed while performing continuous contact with the environment.

The nonlinear Hunt-Crossley model is preferred over the classic Kelvin-Voigt model.

Particular attention is given to the precise recognition of contact between slave robot and environment and to the detection of an occurring deformation.

ACKNOWLEDGEMENT

The work described in this thesis has been realized at Shimono Laboratory of Yokohama National University, based in Yokohama, Japan.

I would like to express my sincere gratitude to my advisor, Professor Roberto Oboe, for giving me the opportunity of experience life as a researcher in a special laboratory and of visiting a fantastic country such as Japan. I would like to thank him for the endless support provided during my staying at YNU, both emotional, moral, professional and economical, and for the subsequent months, and for his enormous patience.

My deepest gratitude goes to my japanese advisor, Professor Tomoyuki Shimono, for his care and support in every part of the project, from the professional part to the burocratic part, to the emotional and moral part. I cannot express enough thanks to him for accepting me into his own laboratory and for allowing me to experience so many aspects of research life in YNU.

Furthermore, I would like to thank my mother Annalisa, for her continuous, unbroken support and encouragement during all my life and for all the sacrifices she has done for my wellbeing, both at home and especially during my period in Japan.
I hope I made you proud.
I love you.

I am grateful to my father Marco for his support throughout the years and for continuously pushing me to do my best in every aspect of my life.

My sincere thanks go to all my friends, both inside and outside university: I would not have lasted a day without you all.

Last but not least, my appreciation is due to all the members of Shimono Laboratory, that had accepted and welcomed me into their life with nothing but happiness and friendship. In particular, I am grateful to Toshiaki Okamoto-san, my tutor in YNU, for helping me in every aspect of life in Japan; to Minoru Yokoyama-san, for being willing to help me professionally whenever I was in need; to Shu-san, Pablo, Goto-san, Tsuboi-san and Hatta-san, for all the help and good memories.

CONTENTS

Appendix

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

The contact between a robot and an object is involved in a number of different tasks performed by robotic systems, such as the grasping and manipulation of objects in various ways [1], [2], [3], [4]. In [1] a robot detects the contact impedance of an object via human demonstration: an operator touches the object with the robotic fingers and data is recorded. The operator has his eyes closed, in order to more closely resemble the robot's condition, relying only on tactile information and not on visual information. These data are then utilized to build a model of the object's workspace, in order to achieve dexterous manipulation. In these cases knowledge of the contact dynamics is fundamental for the success of the operation.
In figure 1.1 is shown the learning procedure via human demonstration for robust grasping and dexterous manipulation.

Environmental characteristics are the key for the implementation of force reproduction techniques, as discussed in [5] and [6]. Using these techniques, a real system can interact with a virtually recreated environment, reproducing the same force response that would derive from the real object.
For instance, in [5] the Bilateral Control technique is utilized to operate two coupled linear motors and to extract information from the environment with which the system is interacting. This particular control technique will be analyzed and implemented in the first section of this thesis.
The extracted data, more precisely force and position informa-

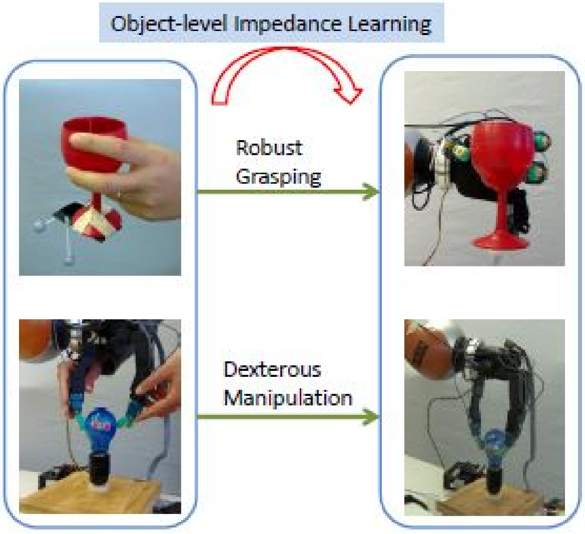


Figure 1.1: Human demonstration for robust grasping and dexterous manipulation

tion, are then utilized to reconstruct a model descriptive of the touched object. Then, using this model it is possible to virtually reproduce the environment behaviour on the operated system. The scheme in figure 1.2 describes this procedure, named Force Sensation Recorder.

This allows the operator to experience the behaviour of a specific object even if the latter is not physically present. The environmental response to the interaction with the system is stored in a *haptic database*, that contains position and force information. The possibilities for the use of this technology are extremely wide, bounded only by limits of imagination.

For instance, it can be utilized for skill education based on force reproduction, as shown in [7] and [8]. This has been proven to be very useful to preserve the memory of all those skills and experiences that are typically passed on from teacher to student. A teacher will execute, using a bilateral system, a procedure of any kind, be it a medical or surgical technique, rather

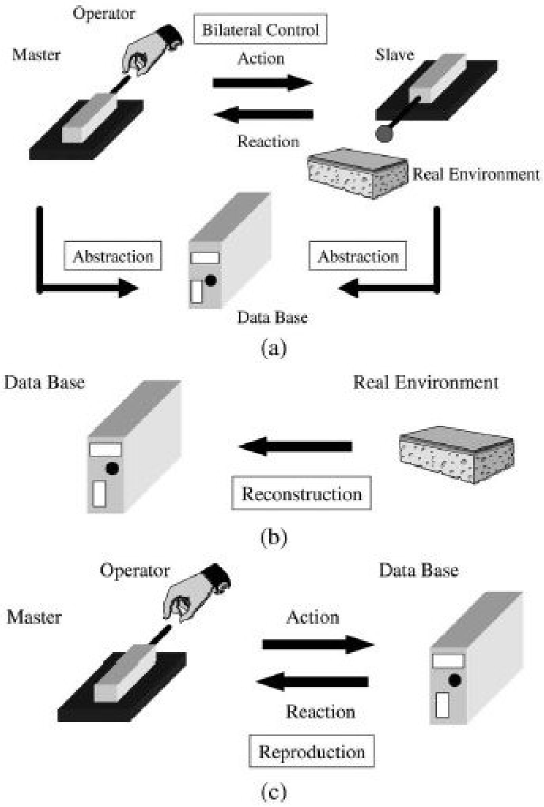


Figure 1.2: Force Sensation Recorder: (a) Data abstraction. (b) Reconstruction of the environment. (c) Virtual reproduction of the environment.

than a specific writing style or any skill whose execution requires a precise technique or to follow fixed steps and stages. The bilateral system stores the information on the procedure in a database, and is then able to reproduce exactly what it has been taught. Furthermore, it is able to teach and guide a student throughout the learning process of that specific procedure. The advantages of this technique become even more evident when thinking about skills that are difficult to replicate, such as particular surgical operations: with the use of the force reproduction technique it is possible for a student to train multiple times on the same operation without the need for a living tissue or organ.

Environmental characteristics are also used for model-based control techniques. One major field of application is the surgical field: as described in [9], minimally invasive surgery can be performed by a robot controlled on the basis of a model of the environment. This allows for safer and more precise motion compared to human operation.

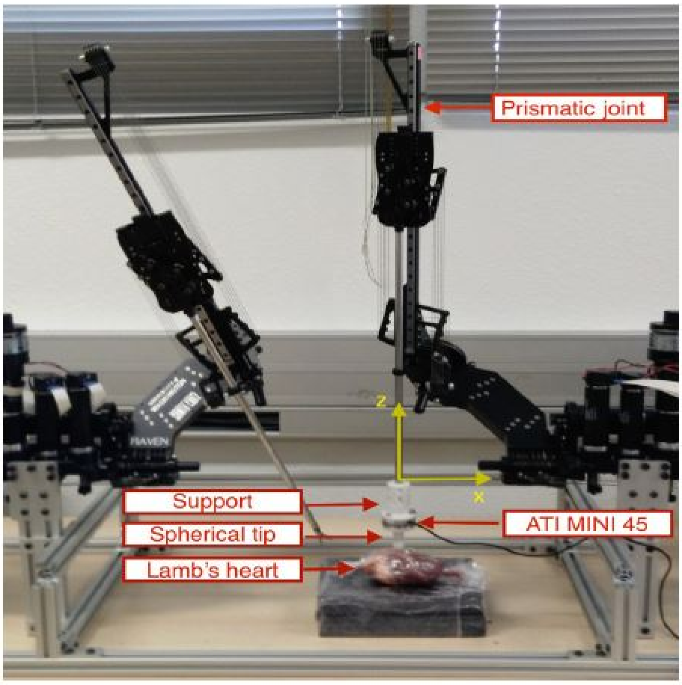


Figure 1.3: Relaxation test performed using the Raven II surgical robot

In [9] a Raven II surgical robot is operated using a force control method based on the Hunt-Crossley model of the environment, combined with a Kalman filter based active observer. This particular model will be further investigated in the following sections of this thesis. A relaxation test is performed to gather information on the tissue contact force-deformation relationship: the system generates a fixed deformation along the direction normal to the surface of contact and measures the reaction force exerted by the tissue. These data are then stored and used off-line to estimate the environment characteristics, that will allow for the reconstruction of the object by means of the Hunt-Crossley model. In figure 1.3 is shown the experimental setup utilized to execute the relaxation test.

Furthermore, knowledge of the environment characteristics allows for the use of Impedance Control [2], [10] or Admittance Control [11], to utilize a motor to simulate the behaviour of a selected environment.

Impedance control is based on the concept that the controller can regulate the dynamic behavior between the robot manipulator motion and the environment [2], rather than considering the motion and force control problems separately. The controlled system will then behave as an object with the selected characteristics. The dynamic behaviour is usually described as an impedance $Z_e(s)$, representing a motion-force relationship as an Ohm law.

$$f(s) = Z_e(s)\dot{x}(s) \tag{1.1}$$

By selecting an impedance value, the control designer specifies the desired dynamic behavior between the motion of the manipulator and the force exerted on the environment.

As stated in [11], impedance control is utilized in many existing systems to reproduce low-stiffness environmental behaviour. However, when the desired dynamics has high-stiffness components, the pole placements of the impedance control systems move towards the imaginary axis. This causes the insurgence of large force vibrations, that can lead to instability of the control system.

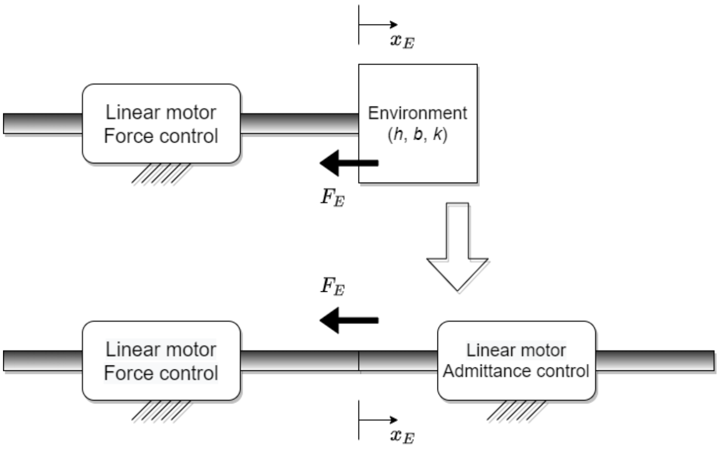


Figure 1.4: Schematic realization of Admittance control

To avoid this problematic situation the Admittance control technique can be utilized. In this case the relationship between system position and reaction force will be the following.

$$\dot{x}(s) = \frac{1}{Z_e(s)} f(s) = Y(s) f(s)$$

. The poles of the controlled system are designed on the real axis, so that no vibration is generated. A schematic representation of the use of Admittance control with linear motors is provided in figure 1.4. In this example a force controlled linear motor interacts with the environment at position x_E, receiving the reaction force F_E from the object. Another motor operated by the admittance control structure can substitute the environment: it will replicate the exact behaviour of the object, exerting the same force F_E at the position x_E.

1.1 IMPEDANCE MODELS

The desired impedance can be modeled in a number of different ways. In [12] is presented an overview of the possible types of transfer function relationship between position and force, summarized in figure 1.5.

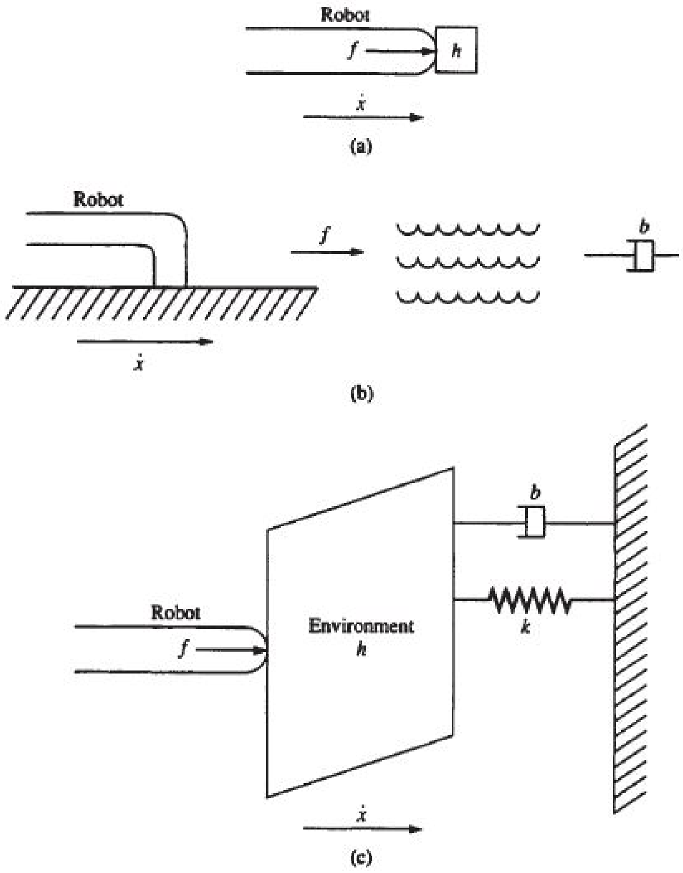


Figure 1.5: Different types of environmental impedance: (a) Inertial impedance. (b) Resistive impedance. (c) Capacitive impedance

- case (a) represents an inertial type of impedance, where the force is proportional to the acceleration of the system, as in the case of a robot moving an object with mass *h*

$$f(s) = h\ddot{x}(s) \tag{1.2}$$

- case (b) describes a resistive environment, depicting a manipulator moving through a liquid substance: the force exerted by the environment is dependent on velocity

$$f(s) = b\dot{x}(s) \tag{1.3}$$

- lastly, in case (c) is represented a complex case, combination of an inertial, resistive and capacitive impedance, assuming the characteristics of the latter: the interaction force is dependent on acceleration, velocity and position, such as in the case of a robot pushing against an object of mass *h*, damping coefficient *b* and spring constant *k*

$$f(s) = h\ddot{x}(s) + b\dot{x}(s) + kx(s) \tag{1.4}$$

Depending on the impedance type, the behaviour of the environment will change drastically. The choice of the environmental characteristics to be taken into account in the analysis will be an integral part of the work described in the following sections of this thesis.

1.2 ESTIMATION PROCEDURE

Information on the environment can be acquired using an estimation algorithm, that processes the response from the environment itself on the robotic actuator. In the conditions considered in this thesis, where measures of force, position and velocity are available, a linear regression algorithm can be implemented.

Regression is a method of modelling a target value based on independent predictors. This method is widely used for defining cause-effect relationship between variables. Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables.

Simple linear regression is a type of regression analysis with a single independent variable and a linear relationship between the independent variable x and dependent variable y. Based on the given data points, the algorithm tries to plot a line that models the points the best. The line can be modelled based on the linear equation shown below.

$$y = a_0 + a_1 x \tag{1.5}$$

The objective of the linear regression algorithm is to find the best values for a_0 and a_1. The output of the algorithm is a graph such as the one shown in figure 1.6.

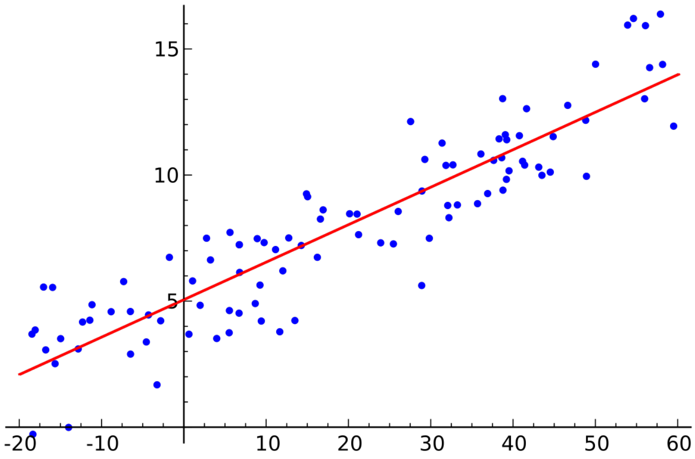


Figure 1.6: Best fitting line (red) for the data points

The most commonly used algorithms in this scenario are based on the Recursive Least Squares structure, in various shape and form [13], [14], [15], [16], [17].

The "least squares" method is a form of mathematical regression analysis used to determine the curve of best fit for a set of data. It aims to create a curve that minimizes the sum of the squares of the errors that are generated by the difference between the results of the associated equations and the target value.

This algorithm can be utilized recursively (RLS), so that each new data point is taken into account to modify a previous estimate of the parameters. In particular, the RLS method allows for the dynamical application of LS to time series acquired in real-time. In this thesis three existing methods for contact impedance estimation are presented and compared.

In particular, the main purpose of this thesis is to achieve the fast online estimation of the environment characteristics: this means that the estimation algorithm should run in parallel with the control structure, requiring a careful optimization of the resources available for the computation.

1.3 CONTACT AND RUPTURE DETECTION

1.3.1 *Contact evaluation*

To achieve a correct estimation of the environment properties it is fundamental to precisely detect the environment position, thus an advanced method for detecting contact is analyzed and applied. The method developed in [18] is based on the use of a *dither*, that is an intentionally applied small noise, into the control loop. This will affect the frequency components of the output, generating harmonics dependent on the dither own pulse.

Thanks to the damping properties of the environment, the amplitude of these components will be highly reduced as soon as the motor comes into contact with the object. Utilizing a time-frequency analysis of the output signal, contact between the robot and the environment can thus be precisely detected.

1.3.2 *Detection of occurring rupture*

Furthermore, unlike a human operator that can rely on experience, a robotic system has no information on how a real object is going to behave upon contact, and this can lead to the deformation and/or damage of said object. As described in [19], when a robot interacts with the environment this will be deformed in three different ways.

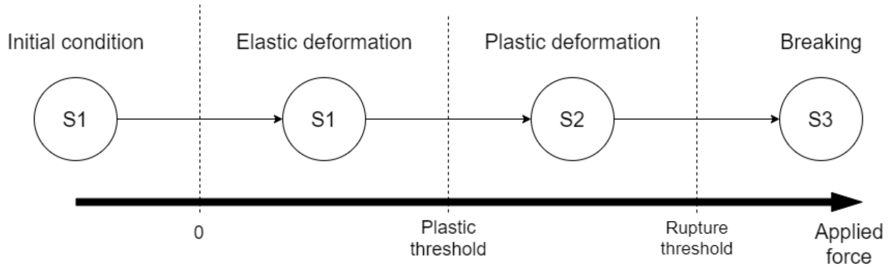


Figure 1.7: Environment state transition in relation to growing applied force

- Elastic deformation: if the applied force is low enough the environment will deform lightly and return to its initial shape when released;
- Plastic deformation: if the applied stress exceeds a limit, dependent on the object properties, the deformation will become permanent and a percentage of it will persist even after release;
- Breaking: applying a force that exceeds the ultimate limit of the material resilience, a rupture will begin spreading, leading to the complete failure of the object.

This is shown in figure 1.7.

The deformation properties are closely related to the stress-strain behaviour of the environment: a typical curve of this type is shown in figure 1.8.

The slope of the curve, constant during the elastic deformation, corresponds to the object stiffness (E in the diagram).

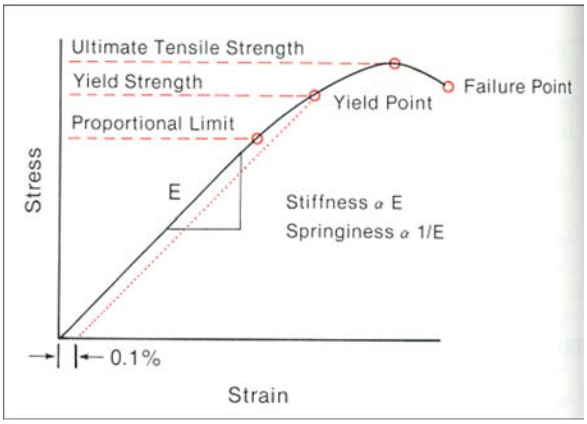


Figure 1.8: Typical stress-strain diagram

When the proportional limit is surpassed, the slope, and consequently the stiffness, decreases up to the ultimate tensile point, where it reaches zero. Utilizing the estimation procedure this variation of the value of the stiffness can be detected: a plastic deformation can then be safely identified and avoided, thus maintaining the integrity of the object.

The thesis is structured as follows: Chapter 2 provides a description of the robotic system and control structure. Then, in Chapter 3 the choice of the environmental model is discussed. In Chapter 4 the description of the estimation algorithms is given. Then in Chapter 5 contact detection is analyzed and the utilized method is evaluated. In Chapter 6 is provided an overview of the method utilized to detect an occurring deformation in the object under evaluation. Lastly, Chapter 7 provides a comparison between simulations and experiments.

2 BILATERAL CONTROL

2.1 INTRODUCTION TO HAPTICS

Robotic systems and intelligent machines are designed and developed to support human activities. Technologies for handling visual and auditory information is today very well established: we have programs that can process these information in the form of text, pictures, audio and video files, etc.
However, apart from sight and sound, humans rely heavily on the sense of touch to execute every kind of practical task. The research field of haptics focuses on this aspect: it examines information deriving from physical interaction between humans and objects. Thanks to this information we can directly identify an object's properties, such as size, shape and location. This is a key element in procedures belonging to the medical field, especially surgical operations, but also for palpation procedures and for grasping and manipulation of delicate objects.

The main goals of haptic systems consist in extending human physical activities and in giving personal support. Robots and machines must be able to sense the environment as we humans do, in order to successfully carry out complex tasks. Moreover, they must be able to transmit to a human operator this type of information.

This is further investigated in [20], where the directional properties of human sensations are described as in figure 2.1.

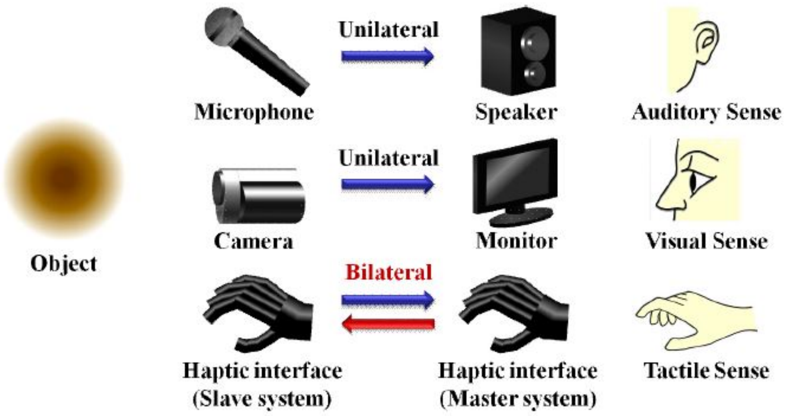


Figure 2.1: Directional properties of human sensations

The auditory information, recorded by a microphone and reproduced by a speaker, and the visual information, recorded by a video camera and reproduced by a monitor or television, are represented as unilateral information flow. Once recorded, the user is able to fully appreciate the information without the need to interact with the source again. This is not true in regard to the haptic information: its intrinsically bilateral nature requires for the presence of a slave system, which the master is going to interact with.

Bilateral Control is a teleoperation technique that allows for the transmission of haptic information between two robotic systems. The term teleoperation or *telefunctioning* [21] refers to a system comprehensive of two robots: one is defined as the "master", operated by a human, and the other, referred to as "slave", executes a task. These two systems are not mechanically connected, and can be placed in remote locations in respect to each other. The slave will move accordingly to the motion given to the master by the operator.

Teleoperation can be defined *unilateral* or *bilateral*, depending on the direction of the signals flowing during operation, as described in [22].

In unilateral teleoperation the information flow is only in one direction. The master system, driven by the human operator, sends the input signals, such as position and/or velocity, through the communications line to drive the slave system. No information is sent back to the master system or the human operator. The slave system has, however, a local closed-loop control system which uses the feedback signals within this control system, to ensure its own stability during operation.

A bilateral teleoperation system is very much the same as in the unilateral case: the teleoperator system consists of the master, the slave and the communication link between them. There is however one important difference: the master system sends position or velocity commands to a slave system, while force or torque information induced from interactions with the environment is fed back to the master by the slave system). This is often referred to as "position-force" architecture, a two-channel architecture with one information flow in each direction. A four-channel type of operation can also be used, in which both position and force signals are transmitted from master to slave system and viceversa.

Master and slave system are in many cases two completely identical systems, but many other possibilities have been explored. The master device can be a 1-DOF joystick, as well as a glove-based interface with many DOF [23]. The slave robotic device may vary from a 1-DOF manipulator to a complex system with a dexterous robot hand attached to a multi degree-of-freedom arm, or even to multiple slave robots [24].

The control architecture necessary to create *telefunctioning* has some interesting and useful properties. It allows the designers to modify and customize the sensibility of the system to uncertain forces, such as the dynamics of the human operator or of the manipulated object, or even the communication delay. Also, being the human in physical contact with the machine, the power transfer between the two is guaranteed. Master and slave robots both have an independent closed-loop controller, to ensure stability even when not in motion and to minimize effects such as friction or non-idealities in the transmission mechanism.

2.2 HISTORY OF BILATERAL CONTROL

Bilateral Control allows a human operator maneuvering the master to vividly sense what the slave is experiencing. This is usually referred as *kinesthetic coupling* [25], [26]: the tactile or *kinesthetic* information exerted from the environment on the slave is fed back to the master and consequently to the human operator.

This control technique has been proven to be useful in a large number of situations. First of all, this type of system can be implemented in those cases in which visual information is absent or non sufficient to perform a desired task, or when an object or a location is not directly accessible for any reason (safety, lack of space, etc.). For instance, this type of system can be useful in presence of a risk for contamination, allowing operators to keep a safe distance while also being able to perform the required tasks. It can also be implemented in case of extreme conditions, such as disaster areas, deep water or even in space:

a robot could reach places potentially dangerous for humans, while being teleoperated from a safe area.

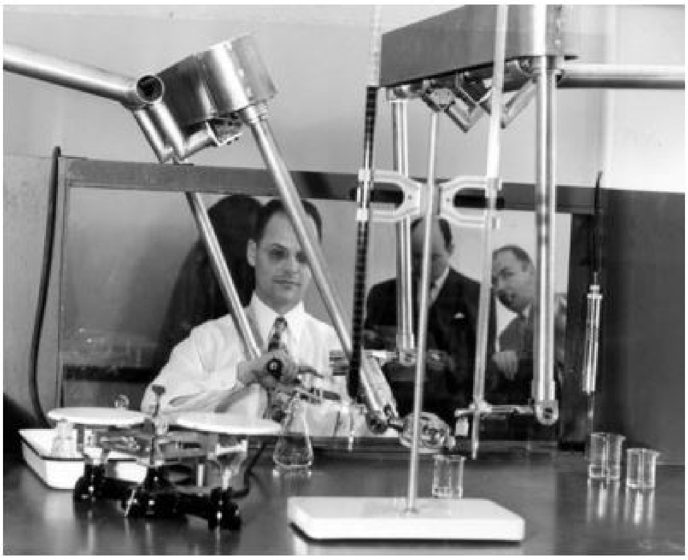


Figure 2.2: Goertz telemanipulator

The first teleoperated manipulator was developed in 1948 by Ray Goertz, at the Argonne National Laboratory (Figure 2.2. It consisted in two systems, referred to as "master" and "slave" system, connected by mechanical linkages and cables. It was developed for the protection of workers from radiation, while enabling precise manipulation of materials, when working on a nuclear hot-cell [27].

In 1950s, the linkage connections were replaced with electric servomotors allowing for a much greater distance between master and slave system.

In 1978 an undersea manipulator for exploration and oil acquisition as developed by Sheridan and Velplank for the Massachussets Institute Of Technology, in the Cambridge Man-Machine Systems Laboratory [28].

Teleoperation was widely used in space applications. In this field, teleoperated systems allow for remote control of activities such as satellite capture and repair.

In 1967 the Surveyor III landed on the surface of the Moon: the operation of the camera mounted on it was completely dependent upon the receipt of proper commands from the Earth [29]. Also on the 1976 Viking spacecraft, which landed on Mars, was mounted a remote controlled manipulator arm, with a collector head, temperature sensor and magnet [30], [31]. Its purpose was to study the biology, chemical composition (organic and inorganic), meteorology, seismology, magnetic properties, appearance, and physical properties of the Martian surface and atmosphere.

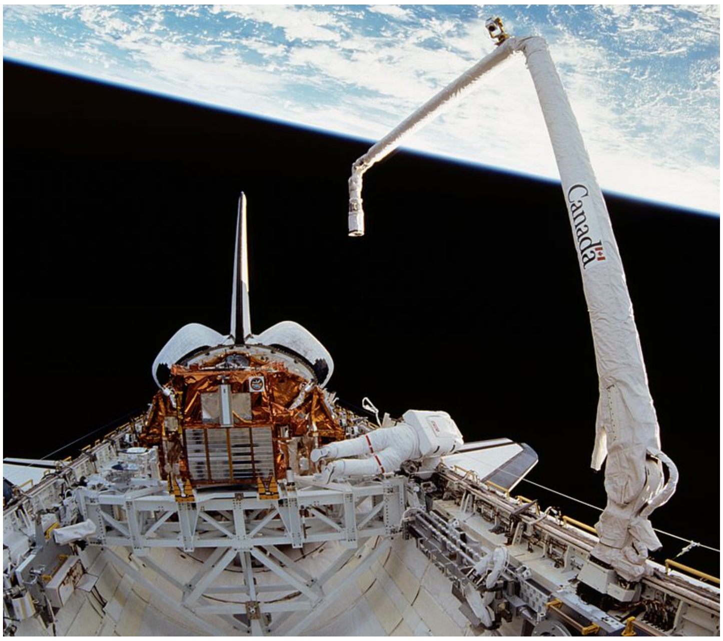


Figure 2.3: The Canadarm mounted on a space shuttle

One of the most important application is surely the Shuttle Remote Manipulator System (SRMS), also named *Canadarm* [32], [33]. Developed by to Spar Aerospace, its function is to deploy, maneuvre, and capture payloads. It is shown in figure 2.3.

For example it was used to retrieve satellites and place them in the cargo bay. It can also be utilized as a mobile platform for astronauts during space walks, as shown in figure 2.4.

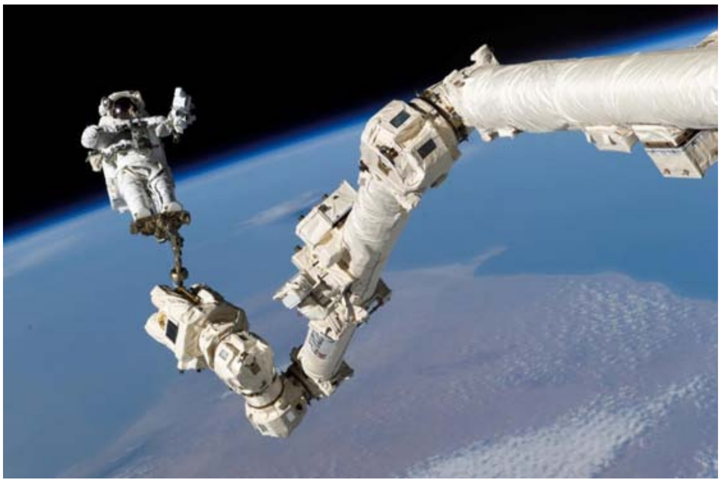


Figure 2.4: An astronaut using a SRMS to work outside the shuttle

NASA has also been working on a system called *Robonaut*, which is a humanoid robot, meant to replace astronauts in dangerous missions, such as space walk, on the space shuttle and/or the space station [34]. NASA began working on the Robonaut project in 1996 and produced the first version of the robot in 2000. It has a head, torso, arms and hands like a person. Vision is provided by cameras in the head and it can perform tasks much like a person utilizing its hands and fingers (figure 2.5): this makes it a dexterous robot. The Robonaut can operate in two different modes:

- it can perform tasks in an automated way: when given a simple order it is capable of computing by itself the operations necessary to fulfill it;

- it can be remotely operated: the operator can use a headset to see what Robonaut sees through its cameras and make it move through its controls.

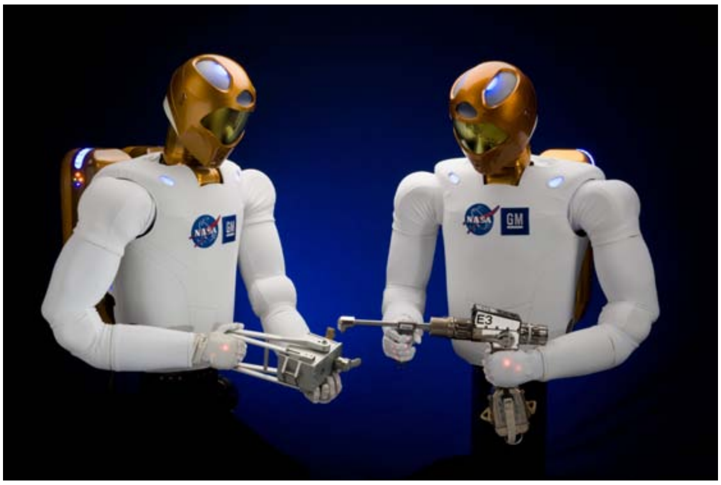


Figure 2.5: Robonauts utilizing tools

In more recent years Japanese researchers have worked on space robot teleoperation technology to achieve effective ground-based control of manual manipulations in orbit [35]. National Institute of Advanced Industrial Science & Technology, [36] Kyoto University of Japan]

The medical field, and specifically the surgical field is another application where remote control is required. Telesurgery and surgical simulations have been widely explored [37], [38] as well as teleoperated nanomanipulation [39].

2.3 ANALYSIS OF TELEMANIPULATION

The condition in which the environmental effects on the slave are transferred to the master without alteration is defined as *telepresence*, referred to as a subclass of telefunctioning [21]. In general, the position signal and/or the haptic feedback signal can be scaled, realizing a different motion of the slave in respect to the master. This allows for macro-micro operations [40]: the motion of the slave can be scaled down, to achieve higher precision, or scaled up to achieve a greater range of motion. Also the two systems can have different sizes: maneuvering a human-sized system can allow for the operation of a much bigger system (e.g. a backhoe) or much smaller system (e.g. a surgical needle).

The use of bilateral control becomes central in the execution of complex tasks that involve articulate and precise movements, such as different types of writing, painting, etc. In relation to these and many more activities it is possible to record the movements of a human expert and then let the robots reproduce them by themselves. This is done with a technique called *Motion Reproduction*: by saving the motion of a teacher in a *haptic database*, a robot can reproduce these movements and perform complex tasks [8], [41], [42].

Motion reproduction can also be utilized in a teaching perspective: a complex operation can be recorded by a teacher and students can utilize the stored data to practice in a virtual environment [5], [7], [8]. This is frequently referred as *skill teletraining* or *skill transfer*. This allows for a number of improvements in the teaching process: for instance the level of difficulty can be adjusted to the current skill level and preparation of the student. Also, students can practise operations that could normally be repeated a finite number of times, or that require precise conditions, preparation and equipment to be performed, such as particular surgical operations.

It is important to notice at this point that although machines can perform tasks autonomously, it is mandatory to design a manual operating mode, as a backup or supplement, in order to allow humans to take control of the robots if the situation so requires.

This control technique is based on the use of two structures:

- the Disturbance Observer (DOB);
- the Reaction Force Observer (RFOB).

The block diagrams of these structures are shown in figure 2.6 [5].

The DOB is capable of detecting the disturbance force in the system without the use of force sensors. It combines information on the ideal characteristics of the system that is being controlled (mass M, torque constant K_t), the input signal (current I_a) and output signal (position x) to extract the force component that is disturbing the correct behaviour of the system. This component can then be compensated to achieve the desired behaviour.

The disturbance force is represented as

$$F_{dis} = F_{ext} + F_{int} + F_C + D\dot{x} + (M - M_n)\ddot{x} + I_a^{ref}(K_{tn} - K_t) \quad (2.1)$$

where

- F_{ext} is the force deriving from the external environment;
- F_{int} represents forces internal to the system;
- F_C is the force from Coulomb friction;
- D is the viscosity constant;
- M and M_n are the real and nominal mass of the system;
- K_t and K_{tn} are the real and nominal torque constant of the system;

- I_a^{ref} is the current reference.

The block diagram of the DOB is shown in figure 2.6(a).

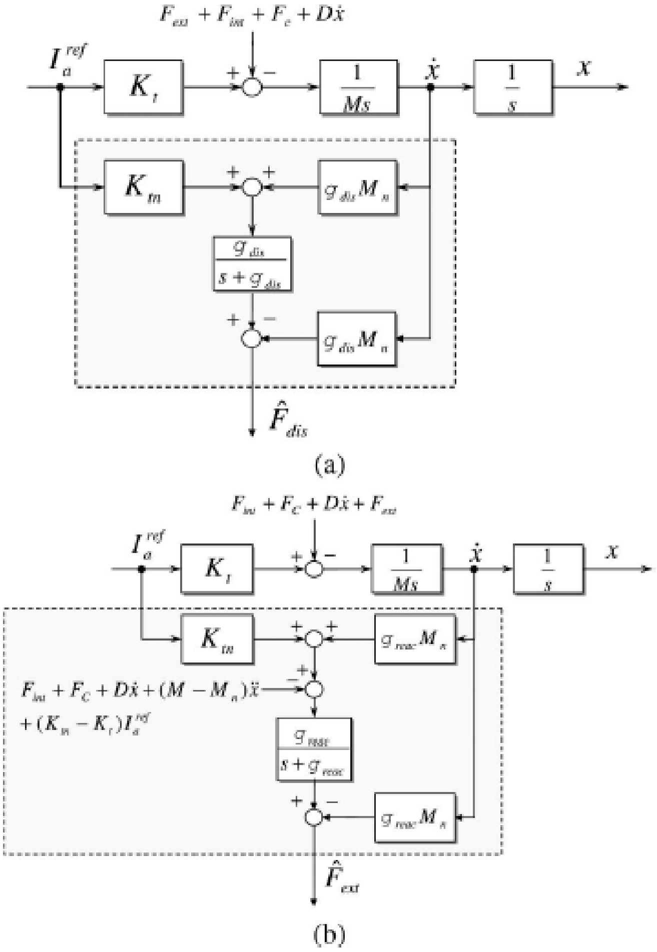


Figure 2.6: Block diagram of a) DOB structure; b) RFOB structure

From equation 2.1 can then be isolated the term relative to the external force F_{ext}: this is the purpose of the RFOB, showed in figure 2.6(b). Previous knowledge of the other disturbance components is required to utilize this structure.

Isolating the external force, corresponding to the force deriving from the manipulated object, is a central component of the work in this thesis. While the DOB structure is used mainly to ensure stability of master and slave system individually, the

RFOB signal will be utilized not only to couple the two motors wih the use of Bilateral Control, but also as the reference signal for the estimation of the properties of the environment.

It is important to remark the fact that no force sensor has been used in this work, thanks to these two structures. This reduces greatly the cost of the experimental setup, while maintaining a good resolution of the force signal. However, due to the presence of a filtered derivative block inside these structures some limitations arise, especially in relation to the precise identification of the time and position of contact between the slave and the environment.

2.5 BILATERAL CONTROL

These techniques allow for the use of bilateral control. The control structure is shown in 2.7.

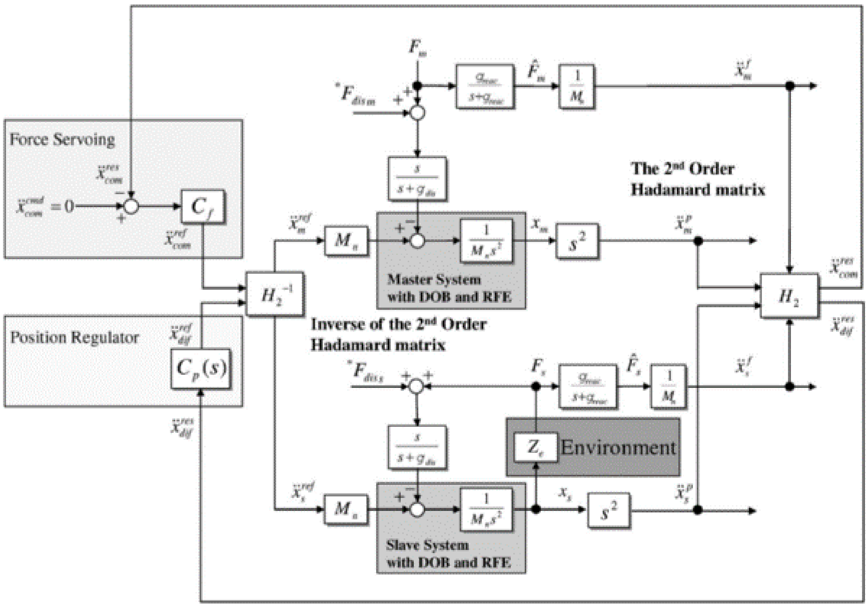


Figure 2.7: Bilateral control structure

This utilizes force and position signals from the system to couple two motors, reproducing the motion of the master on the slave.

The aim of bilateral control can be separated into two objectives:

- *transparency*: artificial realization of the law of action and reaction between the force applied to the master by the operator and the reaction force from the environment to the slave:

$$F_M + F_S = 0 \tag{2.2}$$

- *tracking*: mutual position tracking of the two motors, so that the slave follows perfectly the motion of the master:

$$x_M - x_S = 0 \tag{2.3}$$

These two condition must be satisfied simultaneously for the correct execution of bilateral control.

In [5] is proposed a structure based on acceleration dimension: this allows to calculate equations 2.2 and 2.3 in one common dimension:

$$\begin{cases} \ddot{x}_M^f + \ddot{x}_S^f = \ddot{x}_{com}^{cmd} = 0 \\ \ddot{x}_M^p - \ddot{x}_S^p = \ddot{x}_{dif}^{cmd} \rightarrow 0 \end{cases} \tag{2.4}$$

The first of the 2.4 represents the information from the force dimension, defined as "common mode", while the second denotes information from position dimension, called "differential mode".

In particular, the former is obtained dividing the force signal by the mass of the motor:

$$\ddot{x}_M^f = \frac{F_M}{M_n}, \quad \ddot{x}_S^f = \frac{F_S}{M_n} \tag{2.5}$$

where M_n is used in both cases being the motors identical to each other.

Instead, the differential mode is obtained through second derivation, that implies a delay in the signals. Thus, the use of the arrow sign instead of the equal sign in 2.4.

As can be seen from figure 2.7, position and force signals from master and slave systems are translated into acceleration signals and then combined to obtain the common mode and differential mode responses $\ddot{x}^{res}_{com}, \ddot{x}^{res}_{dif}$. This is done through the use of the second order Hadamard matrix H_2 and its inverse H_2^{-1}:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad H_2^{-1} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.6}$$

Then the force and position controllers are implemented respectively as:

$$\begin{cases} \ddot{x}^{ref}_{com} = C_f(\ddot{x}^{cmd}_{com} - \ddot{x}^{res}_{com}) = K_f(\ddot{x}^{cmd}_{com} - \ddot{x}^{res}_{com}) \\ \ddot{x}^{ref}_{dif} = C_p(s)\ddot{x}^{res}_{dif} = (\frac{1}{s^2}K_p + \frac{1}{s}K_v)\ddot{x}^{res}_{dif} \end{cases} \tag{2.7}$$

where

- K_f is the force gain;
- K_p is the position gain;
- K_v is the speed gain.

Then the acceleration reference signals for master and slave are found using the inverse of the Hadamard matrix H_2^{-1}.

$$\begin{bmatrix} \ddot{x}^{ref}_m \\ \ddot{x}^{ref}_s \end{bmatrix} = H_2^{-1}\begin{bmatrix} \ddot{x}^{ref}_{com} \\ \ddot{x}^{ref}_{dif} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \ddot{x}^{ref}_{com} + \ddot{x}^{ref}_{dif} \\ \ddot{x}^{ref}_{com} - \ddot{x}^{ref}_{dif} \end{bmatrix} \tag{2.8}$$

On both systems are applied the DOB and RFOB structures. The former is used to compensate the disturbances on the motors, identified as:

- disturbances from the power supply, such as a 50Hz oscillation in current;
- friction on the motor shaft;
- discrepancies between the ideal and real values of mass and torque constant;
- reaction force deriving from the environment.

On the other hand, the RFOB structure isolates the external forces, identified as:

- the force applied by the operator on the master F_M;
- the force exerted by the environment on the slave upon contact F_S.

From these signals is extracted the common mode acceleration information using equation 2.5.

In the following chapter will be further investigated how to identify the force from the environment.

3

ENVIRONMENTAL MODEL

To describe the contact force exerted by the environment on the motor a model of the touched object is required.

In the past years many different environmental models have been developed and used. An in depth analysis can be found in [43], [44].

In general two different approaches for contact analysis can be distinguished. The first focuses on short interactions, usually between rigid bodies, where the configuration of the objects does not change significantly. This approach divides the analysis into two intervals, before and after contact, combined with secondary phases, such as slipping, sticking and reverse motion. It is referred to as *discrete* approach, and it is based on the use of various coefficients, mainly the restitution coefficient and the impulse ratio, to describe the process of energy transfer and dissipation.

In order to describe more flexible systems a different approach has to be used. In this case the interaction forces act in a continuous manner during contact between two objects: this allows to perform the analysis of the motion in the usual way, inserting the contact forces to the equation of motion. This approach, referred to as *continuous* approach, is more suitable for describing the real behaviour of the environment, subject to friction and other non-idealities, as well as for describing complex scenarios with multiple contacts and bodies.

3.1 DISCRETE METHODS AND COEFFICIENT OF RESTITUTION

The impact of two bodies causes reaction forces and changes in the velocities of the two objects: this results in elastic and/or plastic deformation, correlated with energy dissipation [43]. The objective of this model is to determine the after-impact conditions of the system, given the initial state.

The dynamic of impact depends on many properties of the bodies in contact, such as material, geometry and velocity. It can generally be separated into two phases, shown in figure 3.1:

- *compression*: it begins at point O, when the two bodies come in contact, and ends at point A, when the maximum deformation is reached, with zero relative velocity;
- *restitution*: from point A the deformation decreases until the moment in which the two bodies separate, at point B, C or D.

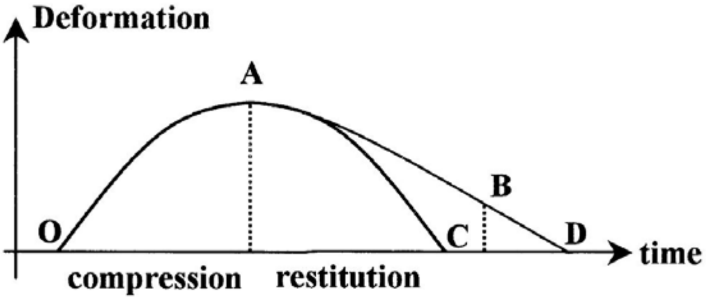


Figure 3.1: Deformation during the impact

Depending on the velocity and the consequent plastic deformation and energy loss, four types of behaviour can be distinguished:

- perfectly elastic impact (line O-A-C): no energy is lost and all of the deformation is retrieved;
- partially elastic impact (line O-A-D): here some energy is lost, but no deformation is permanent;
- partially plastic impact (line O-A-B): both energy loss and permanent deformation;
- perfectly plastic impact (line O-A): all energy is lost and all deformation is permanent.

To express the energy loss during the impact the coefficient of restitution e is used. To satisfy the work-energy principle it should satisfy the condition $0 \leqslant e \leqslant 1$, where:

- $e = 1$ corresponds to the perfectly elastic condition;
- $e = 0$ corresponds to the perfectly plastic condition.

This coefficient depends on many elements, from the geometry and the material of the bodies, to the approach velocity and the duration of contact, to the presence of friction.

3.2 CONTINUOUS APPROACH

For the purposes of this thesis the continuous approach is considered more suitable: the use of the discrete methods is problematic especially in presence of Coulomb friction, attributed to the approximations in the Coulomb model itself, and in relation to the energy conservation during impacts, due to the definition of the coefficient of restitution.

The first continuous model was developed by Hertz in 1882 [45]: it consisted of an elastostatic model, without damping effects. However, the perfectly elastic contact-force model does not account for the energy dissipation in the contact present in mechanical systems. Therefore, it is not possible to model the compression and restitution phase of the contact.

The Kelvin-Voigt model, or spring-dashpot model [46], is the first and most simple representation comprehensive of a damping component: it consists in the parallel of a linear spring and a viscous damper. This model presents some limitations as well: it does not represent the non-linearity of the whole contact process, thus it is only suitable for contacts at higher impact velocities [47].

These two models were then combined into the impact-pair model, presented by Dubowsky and Freudenstein [48]: a viscous damper was paired with a Hertian spring, because of its good representation of the impact surfaces. However this model presents some inconsistencies, due to the dependence of the dissipation component on the contact deformation and the contact velocity [47]. More in depth, this relationship causes the force to have non-zero value at the beginning of the contact and a negative value at the end of the contact process: both phenomenons are physically inconsistent. Furthermore, the coefficient of dissipation remains constant during the whole process, resulting in a constant dissipation, which is again physically inconsistent.

Hunt and Crossley [49] showed that the physical nature of the energy transfer process is not truthfully represented by a linear damping model. The model they then proposed was based on Hertz's theory of contact, combined with a non-linear damping force defined in terms of local penetration. They stated that the exponent of the damping coefficient and of the stiffness coefficient have to be equal.

The hunt-Crossley model was then expanded by other authors, such as Herbert and McWhannell [50], Lee and Wang [51] and Lankarani and Nikravesh [52]. These and other authors focused their work on redefining the hysteresis-damping factor, utilizing different functions to specify the non-linear damping term.

For the purposes of this thesis only the Kelvin-Voigt model and the Hunt-Crossley model will be considered and compared. This choice was made to avoid overcomplication of the model and to keep the computational cost low, while still requiring physical consistency in relation to the real behaviour of the environment.

The two models are described more in detail in this chapter, highlighting the main differences between them, while the estimation algorithms are presented in chapter 4.

3.3 COEFFICIENT OF RESTITUTION

Before studying in detail the different environmental models, it is useful to further investigate the dynamics of the contact, especially the energy loss due to impact. The *coefficient of restitution* e, introduced in section 3.1, is widely utilized to describe this aspect [13]. This coefficient relates the relative velocity of the two objects after the impact v_o with its value at the beginning of the impact v_i:

$$v_o = -ev_i \tag{3.1}$$

In case of perfectly elastic impacts the coefficient of restitution has value $e = 1$ and no energy is lost. In experimental cases, conducted with low impact velocity within the elastic range of the material, e shows a dependence on the initial velocity v_i:

$$e = 1 - \alpha v_i \tag{3.2}$$

with the coefficient α depending on the materials of the two bodies, usually $0.08 \leqslant \alpha \leqslant 0.32 s/min$.

Using these relations, the energy ΔH dissipated in the impact of an object with mass m and initial momentum $p_i = mv_i$ can be expressed as

$$\Delta H = \frac{p_i^2}{2m}(1 - e^2) \approx \alpha v_i \frac{p_i^2}{m} \tag{3.3}$$

The dissipated energy can also be described with a graphic approach. Figure 3.2 shows the *hysteresis curve*, or *stress-strain curve*, generated within the impact between a rigid tool and a laryngeal muscle, as described in [9].

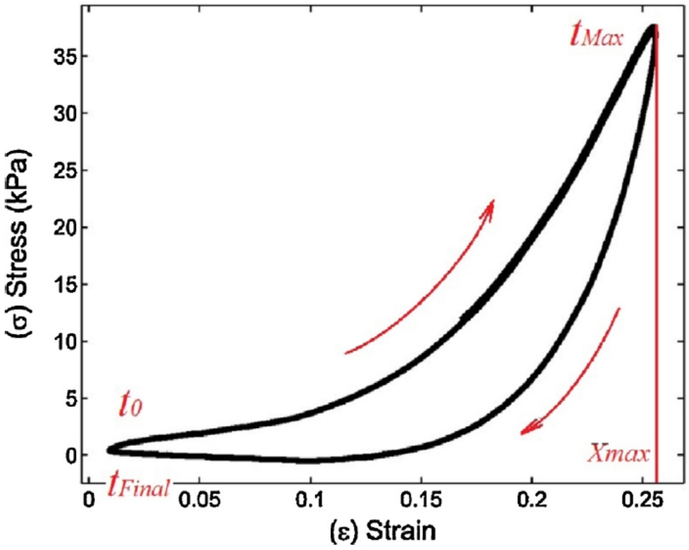


Figure 3.2: Stress-strain curve for a section of laryngeal muscle

In this graph it is simple to distinguish the two phases of the impact introduced in section 3.1:

- the compression, or loading, phase corresponds to the upper part of the curve, going from time t_0 to t_{Max},

- while the restitution, or unloading, phase is represented by the lower part of the curve, between t_{Max} and t_{final}

The power flow during contact can be calculated as

$$P(t) = F_e(t)\dot{x}(t) \tag{3.4}$$

The dissipated energy ΔH is represented, for linear viscoelastic models, by the area of the enclosed loop. It can be calculated as the algebraic sum of the energy corresponding to the compression phase and to the restitution phase, both obtained as the integral of the power flow 3.4 over the respective time intervals.

This analysis will be conducted for both the Kelvin-voigt model and the Hunt-Crossley model in the following sections, highlighting some of the problems and physical inconsistencies concerning the use of the first method.

3.4 KELVIN-VOIGT MODEL

The simplest and most used model is arguably the Kelvin-Voigt model, in which the environment is represented as the parallel of an ideal linear spring and a viscous damper, as shown in figure 3.3.

The parameters used to describe the environment properties are the stiffness coefficient K and the viscosity coefficient D.

$$F(t) = \begin{cases} Kx(t) + D\dot{x}(t), & x \geqslant 0 \\ 0, & x < 0 \end{cases} \tag{3.5}$$

The advantages of this particular model reside in its simplicity and fairly high consistency with the behaviour of the real en-

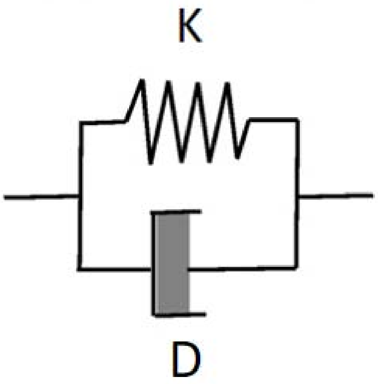


Figure 3.3: Kelvin-Voigt environmental model

vironment. However, some incongruity arise when describing the behaviour of soft materials, with highly viscous behaviour.

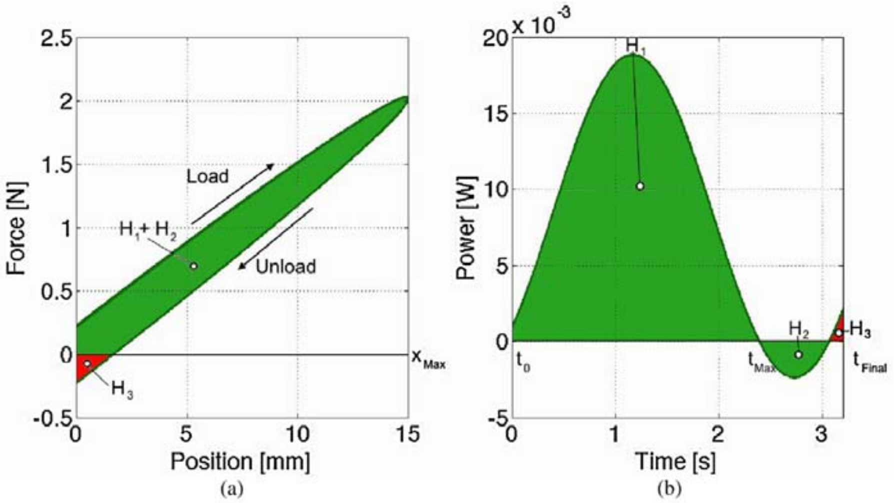


Figure 3.4: Behaviour of Kelvin-Voigt model upon contact: a) Hysteresis loop, b) Power exchange

As discussed in [13], the hysteresis loop obtained simulating the impact of a mass m with the material modeled in equation (3.5) is showed in figure 3.4(a) [9]. The upper arc corresponds to the compression phase, when the environment is deformed from the initial condition $x = 0$ to the final state $x = x_M$, while the lower arc corresponds to the restitution phase, comprehensive of the removal of stress and the return to the initial position.

Due to unnatural shock forces upon impact (time t_0) and tensile forces upon load removal (time t_{Final}), physical inconsistencies emerge in the model behaviour. As showed in 3.4(b), during compression the energy H_1 is flowing from the mass to the object, while the energy flowing during restitution is represented by the sum of areas H_2 and H_3. The total dissipated energy ΔH is then computed as the algebraic sum of the energies H_1, H_2 and H_3:

$$\Delta H_{KV} = H_1 - H_2 + H_3 \tag{3.6}$$

In particular, H_3 represents power flowing, again, from the mass to the touched material during the restitution phase: this is in conflict with both intuition and experimental results.

Furthermore, as shown in various studies ([13], [43], [53]), from a model such as the one in (3.5) can be extracted an equivalent coefficient of restitution e. In the case of the impact of a mass m with an object represented with the Kelvin-Voigt model, e is expressed as:

$$e = \mathrm{e}^{-D\pi/\sqrt{4mK-D^2}} \tag{3.7}$$

This shows no dependence of the coefficient of restitution with the initial impact velocity v_i: this is in contrast with the conditions introduced in section 3.3. This is correlated with the energetic inconsistencies presented in this section.

This issue can be solved by making the viscous force dependent on the penetration depth x, as in the Hunt-Crossley model:

$$F(t) = \begin{cases} kx^n(t) + \lambda x^n(t)\dot{x}(t), & x \geqslant 0 \\ 0, & x < 0 \end{cases} \tag{3.8}$$

The scheme for the Hunt-Crossley model can be extracted from the one in figure 3.3, expressing the Kelvin-Voigt coefficients as:

$$\begin{cases} K = kx^{n-1}(t) \\ D = \lambda x^n(t) \end{cases} \tag{3.9}$$

The exponent n is a real number, usually $1 < n < 2$, and it varies according to the geometry of contact surfaces.

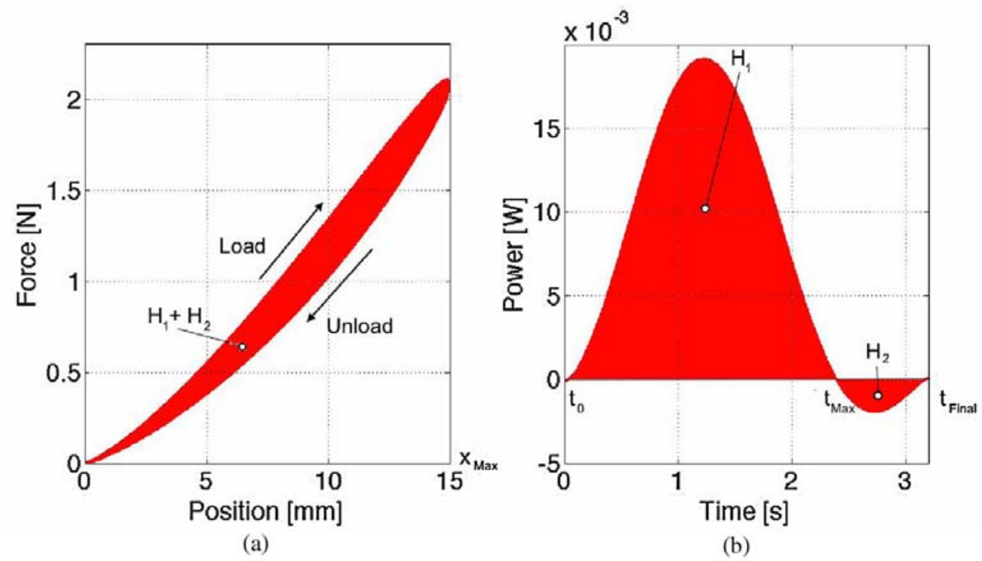


Figure 3.5: Behaviour of Hunt-Crossley model upon contact: a) Hysteresis loop, b) Power exchange

The new hysteresis loop is showed in 3.5. The cycle is correctly closed on itself, while the power exchange between the two objects is divided in two phases: it flows from the mass m to the environment during the compression phase, and vice-versa during the restitution phase. The dissipated energy ΔH is calculated as:

$$\Delta H_{HC} = H_1 - H_2 \tag{3.10}$$

It can easily be noticed that there is no flow of power from the mass m to the environment during the restitution phase, as happened in the previous case. So, the behaviour of this model is proven to be more consistent with known experimental results.

Furthermore, studies have proved that the Hunt-Crossley model accuracy at low speed is higher than the Kelvin-Voigt model [13], [9].

As stated in section 3.4, it is possible to extract an equivalent coefficient of restitution also from the model in 3.8, through its relationship with the stiffness and damping coefficients [13], obtaining:

$$e = 1 - \frac{2}{3}\frac{\lambda}{k}v_i \tag{3.11}$$

In the case of the Hunt-Crossley model the coefficient e is dependent on the initial impact velocity v_i, coherently with the analysis presented in section 3.3.

Furthermore, it can be noticed that the exponent n does not affect the value of e, meaning that the shape and dimension of the contact surfaces have no effect on the behaviour of the objects during the restitution phase.

3.6 CASE STUDY

The experiments performed in this work consider only one-dimensional deformation: the motors utilized are capable of moving linearly in one direction, while the end-effector is a metallic tip. This allows to consider the contact as single-point. Figure 3.6 shows the environment model applied to the system in figure 7.1.

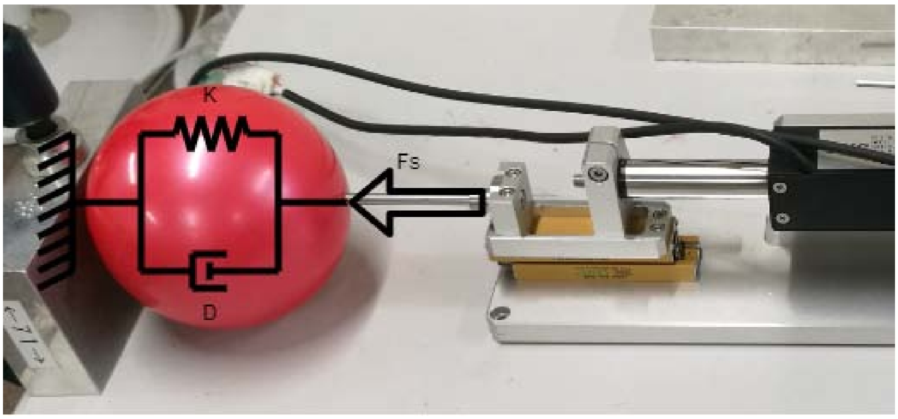


Figure 3.6: Environmental model applied to the system

With the use of these models it is possible to draw a connection between the force signal acquired from the slave RFOB, corresponding to the force exerted by the environment, and the properties of the environment itself.
Depending on the model considered the estimated characteristics will be different: in the following chapter the procedure for parameter estimation is investigated for the two models and the results are compared.

4 ESTIMATION ALGORITHM

Utilizing the theories presented in the previous chapter it is now possible to represent an unknown environment, with which the controlled system comes into contact. We then need a method to extract the environmental characteristics from the information received from the object.

In the system used in this thesis, discussed in chapter 2, the control structure receives the position information from the encoders mounted on both master and slave system. These signals can then be processed with a filtered derivative to obtain velocity and acceleration signals. Then the force signals are received from the RFOB structures on the two systems.

Under these conditions a linear regression algorithm can be implemented.

4.1 INTRODUCTION ON LEAST-SQUARES ALGORITHMS

As briefly discussed in the introduction of this thesis, regression is a method of modelling a target value based on independent predictors, utilized mainly for defining cause-effect relationship between variables.

The *Least-Squares* (LS) method is a form of mathematical regression analysis used to determine the curve of best fit for a set of data. It aims to create a curve that minimizes the sum of the squares of the errors that are generated by the difference between the results of the associated equations and the target value.

This objective is represented as the minimization of a cost function J, that generally assumes the form:

$$J = \frac{1}{n}\sum_{i=1}^{n}(y_i^{ref} - y_i)^2 \tag{4.1}$$

The term y_i^{ref} corresponds to the target value, or predicted value, while y_i is the value reconstructed by the algorithm, that in the case of a linear LS algorithm is calculated as:

$$y_i = a_0 + a_1 x_i \tag{4.2}$$

The difference between these two terms gives the *estimation error*

$$\epsilon_i = y_i^{ref} - y_i \tag{4.3}$$

Then the cost function J represents the average squared error over all the data points, also regarded as the Mean Squared Error (MSE) function.

4.2 RECURSIVE LEAST-SQUARES ALGORITHM

The Least-Squares algorithm can be utilized recursively (RLS), so that each new data point is taken into account to modify the previous estimate of the parameters. In particular, the RLS method allows for the dynamical application of LS to time series acquired in real-time.

This method is analyzed extensively by Lyung and Soderstrom in [54].

This method is a key instrument for the use of advanced control techniques such as adaptive control, adaptive filtering, adaptive signal-processing problems, et al.

The use of the RLS algorithm has some advantages over the classic LS method:

- as already stated, it can be utilized online, allowing for the use of various control techniques and applications;
- thanks to this feature, there is no need to keep the old data, resulting in less memory usage; also, the online update procedure can be stopped based on the estimate of the accuracy of the current model, provided by the algorithm itself;
- this method can also process a batch of data offline, resulting in an more efficient alternative to conventional offline procedures.

The Recursive Least-Square algorithm presents also some disadvantages in contrast with the offline procedure:

- the algorithm needs a model to work with, so this choice must be made *a priori*;
- the accuracy of the RLS is usually not as good as offline method, but it is close enough for long data records.

4.3 INITIALIZATION

Now we need a way to implement the RLS algorithm in the code. The process of initialization presented in this section is extensively described in [54].

Let us consider a dynamical system with input signal $u(t)$ and output signal $y(t)$, sampled in discrete time $t = 1, 2, 3, ...$.

Supposing that the sample values are linearly dependent on each other, this relationship can be expressed as:

$$\begin{aligned} y(t) + a_1 y(t-1) + ... + a_n y(t-n) = \\ = b_1 u(t-1) + ... + b_m u(t-m) + v(t) \end{aligned} \tag{4.4}$$

where

- m is the dimension of the input signal $u(t)$;
- n is the dimension of the output signal $y(t)$;
- $v(t)$ is a disturbance of any kind.

We can represent this equation with a vectorial approach, gathering together all the parameters a_i, b_i in a column vector θ, and all the lagged input-output data into another column vector ψ:

$$\begin{cases} \theta = [a_1, ..., a_n, b_1, ..., b_m] \\ \psi(t) = [-y(t-1), ..., -y(t-n), u(t-1), ..., u(t-m)] \end{cases} \tag{4.5}$$

Then the relationship in (4.4) can be rewritten as:

$$y(t) = \theta^T \psi(t) + v(t) \tag{4.6}$$

that describes the observed variable $y(t)$ as the sum of an unknown linear combination of the components of the observed vector $\psi(t)$ and some noise $v(t)$.

This model is very common in statistics and it is called a *linear regression*, while the components of $\psi(t)$ are called *regression variables* or simply *regressors*.

The parameter vector θ is estimated by the algorithm from measurements of $y(t)$ and of $\psi(t)$, $t = 1, 2, 3, ..., N$, usually minimizing the *estimation error* $v(t)$:

$$v(t) = y(t) - \theta^T \psi(t) \tag{4.7}$$

This component represents everything that is left unexplained by the model: minimizing it means reaching a complete and correct description of the phenomenon object of the study.

Then the cost function is expressed as:

$$J(\theta) = \frac{1}{N}\sum_{t=1}^{N}[\nu(t)]^2 = \frac{1}{N}\sum_{t=1}^{N}[y(t) - \theta^T\psi(t)]^2 \tag{4.8}$$

This function can be minimized analytically with respect to θ, obtaining:

$$\frac{dJ}{d\theta}(\theta) = 0 \Rightarrow \hat{\theta}(N) = \left[\sum_{t=1}^{N}\psi(t)\psi^T(t)\right]^{-1}\sum_{t=1}^{N}\psi(t)\psi^T(t) \tag{4.9}$$

This expression can be simplified by defining a matrix $\overline{R}(t)$:

$$\overline{R}(k) = \sum_{k=1}^{N}\psi(k)\psi^T(k) \tag{4.10}$$

Substituting (4.10) into (4.9), and with the use of some algebra, we obtain:

$$\begin{aligned} &\hat{\theta}(t) = \hat{\theta}(t-1) + \overline{R}^{-1}(t)\psi(t)\left[y(t) - \hat{\theta}^T(t-1)\psi(t)\right] \\ &\overline{R}(t) = \overline{R}(t-1) + \psi(t)\psi^T(t) \end{aligned} \tag{4.11}$$

Sometimes it is easier to work with a different matrix:

$$R(t) = \frac{1}{t}\overline{R}(t) \tag{4.12}$$

which, when substituted into the (4.11), gives a first representation of the *Recursive Least-Squares* algorithm:

$$\begin{cases} \hat{\theta}(t) = \hat{\theta}(t-1) + \frac{1}{t}R^{-1}(t)\psi(t)\left[y(t) - \hat{\theta}^T(t-1)\psi(t)\right] \\ R(t) = R(t-1) + \frac{1}{t}\left[\psi(t)\psi^T(t) - R(t-1)\right] \end{cases} \tag{4.13}$$

4.3.1 *Reduction of computational cost*

The algorithm in (4.13) is coherent with the definition of the RLS method given previously in this chapter, but it is not well suited for computation, since at every new step the inverse of the matrix $R(t)$ has to be calculated. A new formulation with less computational cost can be found introducing a new matrix, named *covariance matrix*:

$$P(t) = \overline{R}^{-1}(t) = \frac{1}{t}R^{-1}(t) \tag{4.14}$$

We then make use of the *matrix inversion lemma*, that states: being A, B, C and D be matrices of compatible dimensions, so that the product BCD and the sum $A + BCD$ exist, the following relationship is true:

$$[A + BCD] = A^{-1} - A^{-1}B\left[DA^{-1}B + C^{-1}\right]^{-1}D \tag{4.15}$$

Applying (4.15) to (4.14), we obtain:

$$\begin{aligned} P(t) &= \left[P^{-1}(t-1) + \psi(t)\psi^T(t)\right]^{-1} \\ &= P(t-1) - \frac{P(t-1)\psi(t)\psi^T(t)P(t-1)}{1+\psi^T(t)P(t-1)\psi(t)} \end{aligned} \tag{4.16}$$

From (4.16) we also find that:

$$\frac{1}{t}R^{-1}(t)\psi(t) = P(t)\psi(t) = \frac{P(t-1)\psi(t)}{1+\psi^T(t)P(t-1)\psi(t)} \tag{4.17}$$

This is defined as a matrix $L(t)$, named *gain matrix*.

Finally, the complete structure of the algorithm is the following.

$$\begin{cases} \hat{\theta}_(t) = \hat{\theta}(t-1) + L(t)\left[y(t) - \hat{\theta}^T(t-1)\psi(t)\right] \\ L(t) = \frac{P(t-1)\psi(t)}{1+\psi^T(t)P(t-1)\psi(t)} \\ P(t) = P(t-1) - \frac{P(t-1)\psi(t)\psi^T(t)P(t-1)}{1+\psi^T(t)P(t-1)\psi(t)} \end{cases} \tag{4.18}$$

This is usually known as the standard *Recursive Least-Squares algorithm*, as it is the most widely used, thanks to its robustness and easy implementation.

This structure has been modified and improved many times by a lot of different authors, with different specific purposes. In the case of this thesis the standard structure, as presented in (4.18, has been chosen, to keep the computational cost contained and to benefit of the aforementioned advantages.

Before diving into the implementation of the algorithm in the specific case study of this thesis, it is necessary to clarify some aspects that influence the correct execution of the algorithm, in particular the setting of the initial conditions.

4.3.2 *Initial conditions*

A recursive algorithm needs some initial values to start the estimation. In the case of the algorithm in (4.18) the values $\hat{\theta}(0)$ and $P(0)$ are needed.

Since $P(t)$ is derived from the matrix $\overline{R(t)}$, under the assumption that this is invertible, the initial value is found at the time t_0 when $\overline{R(t_0)}$ first becomes invertible. Typically this instant of time is calculated as $t_0 = \dim \psi(t) = \dim \theta$ The proper initial values are then found as:

$$\begin{aligned} P(t_0) &= \left[\sum_{k=1}^{t_0} \psi(k)\psi^T(k) \right]^{-1} \\ \hat{\theta}(t_0) &= P(t_0) \sum_{k=1}^{t_0} \psi(k)y(k) \end{aligned} \tag{4.19}$$

However, it is common to adopt a less strict approach and start the estimation at time $t = 0$ with an invertible matrix $P(0)$ and a vector $\hat{\theta}(0)$, obtaining the estimates

$$\hat{\theta}(t) = \left[P^{-1}(0) + \sum_{k=1}^{t} \psi(k)\psi^T(k) \right]^{-1} \left[P^{-1}(0)\hat{\theta}(0) + \sum_{k=1}^{t} \psi(k)y(k) \right] \tag{4.20}$$

The relative importance of the initial values decays over time, as the magnitude of the sums increases. Therefore, a common choice of initial values is:

$$\begin{cases} P(0) = c \cdot I, \quad c >> 0 \\ \hat{\theta}(0) = 0 \end{cases} \tag{4.21}$$

4.4 FORGETTING FACTOR

Another element that can be inserted in the recursive algorithm is the *Forgetting Factor* β: this component modifies the influence of the new samples on the output and on the tracking performances and the convergence rate.

It can be used in a number of different ways. The simplest use of the forgetting factor is to set it to a constant value, usually between 0.98 and 1, to improve the tracking performances. However this prevents the algorithm from reaching convergence, because every variation in the input signal is weighted more than the past samples.

Instead, the use of a dynamically updated forgetting factor can substantially improve the performances of the estimation algorithm. The value of β will then be modified depending on different factors, to vary the influence of the measures on the estimation:

- when $\beta = 1$ the new entries have the same relevance as the older ones, thus enhancing the robustness of the algorithm to noise;

- when $\beta < 1$ the weight related to the older samples decays, in order to achieve fast tracking performances during transients.

In literature can be found numerous different ways of updating β. In this work we focus on two types: the *Exponentially Weighted FF* and the *Error Dependent FF*.

4.4.1 *EWRLS*

One simple update method is the Exponentially Weighted RLS algorithm.

$$\beta = 1 - \alpha e^{-\mu t} \tag{4.22}$$

The forgetting factor is updated at the moment of contact ($t = 0s$) and gradually increases up to 1. This way the algorithm rapidly reaches convergence and gains robustness to noises.

The downside to this approach is that the updating law is not dependent on the actual condition of the algorithm, but only on time. Hence, the performances are not optimal.

4.4.2 *Error-dependent forgetting factor*

In [13] *Diolaiti et al.* utilized an update law based on the analysis of the estimation error, as shown in equation (4.23).

$$\beta = 1 - \alpha_1 \left\{ \frac{1}{\pi} \mathrm{Atan} \left[\alpha_2 \left(|F_{DOB}(h) - \hat{F}(h)| - \alpha_3 \right) \right] + \frac{1}{2} \right\} \tag{4.23}$$

where

- $\epsilon = F_{DOB}(h) - \hat{F}(h)$ is the force estimation error
- $1 - \alpha_1$ is the value of β when ϵ is large
- α_2 influences the transition region
- α_3 is the threshold between small and large error

When the force estimation error is greater than the threshold the forgetting factor decreases up to the value $1 - \alpha_1$ to increase the speed of estimation, while grows up to 1 when ϵ is low.

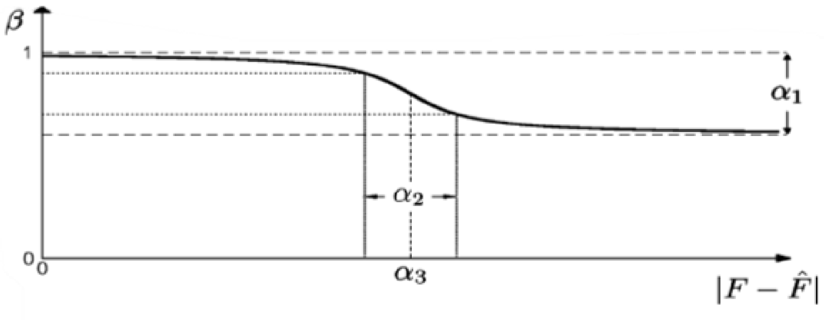


Figure 4.1: Update law of forgetting factor β

The influence of the parameters $\alpha_1, \alpha_2, \alpha_3$ on the forgetting factor $\beta(\epsilon)$ is shown in figure 4.1.

This allows the algorithm to rapidly detect changes in the reference values.

4.5 IMPLEMENTATION OF THE RLS ALGORITHM

Now that all the required elements have been presented, we can proceed to implement the RLS algorithm in relation to the environmental model. Depending on the chosen model, the components of the algorithm will change.

The structure of the estimator Γ utilized in this thesis is shown as equation (4.24):

$$\Gamma : \begin{cases} \hat{\theta}_{t+1} = \hat{\theta}_t + L_{t+1}(y_{t+1} - \psi_{t+1}^T \hat{\theta}_t) \\ L_{t+1} = \frac{P_t \psi_{t+1}}{\beta + \psi_{t+1}^T P_t \psi_{t+1}} \\ P_{t+1} = \frac{[\mathbb{I} - L_{t+1} \psi_{t+1}^T] P_t}{\beta} \end{cases} \tag{4.24}$$

where

- $\hat{\theta}$ are the estimated parameters;
- ψ are the input signals;
- y is the system output;
- β is the forgetting factor;
- L is the gain matrix;
- P is the covariance matrix.

By analyzing the estimation error

$$\hat{e} = y_{t+1} - \psi_{t+1}^{T}\hat{\theta}_{t} \tag{4.25}$$

the parameters $\hat{\theta}$ are modified until convergence is reached, corresponding to the condition $\hat{e} \to 0$.

Depending on various choices, such as the environmental model or the constitutive equation, the elements of these vectors may change drastically. In the following sections three different implementation of the RLS estimation algorithm are described.

4.5.1 *Single-stage estimation method*

At first the Kelvin-Voigt model has been used: based on equation 3.5, the algorithm will derive the stiffness constant K and the viscous constant D from the force signal F_{RFOB}.
The elements in the algorithm are defined as follows:

$$\hat{\theta} = \begin{bmatrix} \hat{K} \\ \hat{D} \end{bmatrix}, \quad \psi = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \quad y = F_{RFOB} \tag{4.26}$$

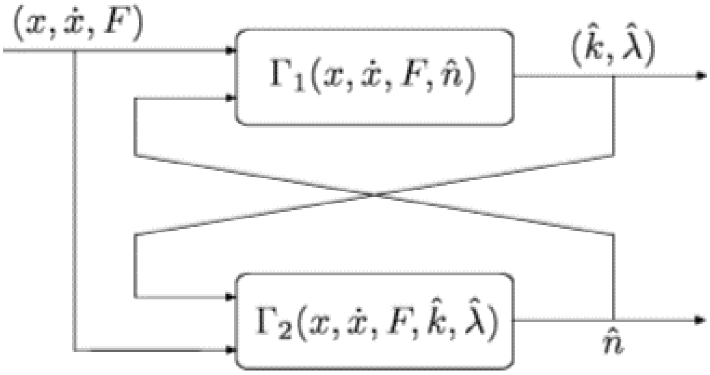
In this case the dimension of the system is 2:

$$Q \in \mathbb{R}^{2x1}, R \in \mathbb{R}^{2x2}$$

4.5.2 *Double-stage estimation method*

For the reasons explained in section 3 the Hunt-Crossley model has then been used. The parameters to be estimated are now the stiffness constant k, the viscous constant λ and the exponent n, as described in equation 3.8.

In [13] *Diolaiti et al.* proposed an estimation algorithm obtained by the combination of two RLS structures connected in mutual feedback (Figure 4.2).



$(x, \dot{x}, F)$
$\Gamma_1(x, \dot{x}, F, \hat{n})$
$(\hat{k}, \hat{\lambda})$
$\Gamma_2(x, \dot{x}, F, \hat{k}, \hat{\lambda})$
$\hat{n}$

Figure 4.2: Double-stage estimator with feedback connection

The elements in the algorithm are defined as follows:

$$\begin{aligned}
\Gamma_1: \quad & \hat{\theta}_1 = \begin{bmatrix} \hat{k} \\ \hat{\lambda} \end{bmatrix}, \quad \psi_1 = \begin{bmatrix} x^n \\ x^n \dot{x} \end{bmatrix}, \quad y_1 = F_{RFOB} \\
\Gamma_2: \quad & \hat{\theta}_2 = \hat{n}, \quad \psi_2 = \log(x), \quad y_2 = \log\left(\frac{F_{RFOB}}{k + \lambda\dot{x}}\right)
\end{aligned} \tag{4.27}$$

The feedback connection of the two estimators Γ_1 and Γ_2 influences the convergence of the overall procedure. To achieve convergence it is necessary that the disturbances derived from the interconnection do not bias the estimation error of Γ_1 and Γ_2. In other words, the estimator Γ_2 should converge independently from Γ_1.

This condition can be formulated as

$$\|\delta k + \delta\lambda\dot{x}\| \ll \left\|\frac{\epsilon_1}{x^n}\right\|, \quad \forall h = 0, 1, 2, \ldots \tag{4.28}$$

where

- $\delta k = \hat{k} - k$
- $\delta\lambda = \hat{\lambda} - \lambda$
- $\epsilon_1 = F - (\hat{k} + \hat{\lambda}\dot{x})x^{\hat{n}}$

Furthermore, $\delta n = \hat{n} - n$ is assumed to be always small, in order to validate the approximation

$$1 - x^{\delta n} \simeq -\delta n \log x \tag{4.29}$$

These conditions are not always satisfied during the estimation process and this may compromise the quality of the estimation.

4.5.3 *Single-stage logarithmic method*

The limitations presented in section 4.5.2 can be avoided using a different approach to the linearization of equation 3.8.

Presented by *Haddadi* & *Hashtrudi-Zaad* in [14], the logarithmic approach allows to estimate all three parameters of the Hunt-Crossley model in a single stage.

This method considers the logarithm of both elements in equation 3.8 to obtain:

$$\begin{aligned} \ln[F_{RFOB}(t)] &= \ln\left[kx^n(t)\left(1 + \frac{\lambda\dot{x}(t)}{k} + \frac{\epsilon}{kx^n(t)}\right)\right] \\ &\approx \ln(k) + \frac{\lambda}{k}x\dot{(}t) + n\ln[x(t)] + \frac{\epsilon}{kx^n(t)} \end{aligned} \tag{4.30}$$

where ϵ represents both modelling error and measured noise during the process.

The elements in the algorithm are now defined as follows:

$$\hat{\theta} = \begin{bmatrix} \ln(\hat{k}) \\ \frac{\hat{\lambda}}{\hat{k}} \\ \hat{n} \end{bmatrix}, \quad \psi = \begin{bmatrix} 1 \\ \dot{x} \\ \ln(x) \end{bmatrix}, \quad y = \ln(F_{RFOB}) \tag{4.31}$$

and the estimated parameters are then derived as:

$$\hat{k} = e^{\hat{\theta}(1)}, \quad \hat{\lambda} = e^{\hat{\theta}(1)}\hat{\theta}(2), \quad \hat{n} = \hat{\theta}(3) \tag{4.32}$$

The dimension of the system increases to 3:

$$Q \in \mathbb{R}^{3x1}, R \in \mathbb{R}^{3x3}$$

However, some conditions need to be satisfied in order for this method to be validated:

$$\left|\frac{\hat{\lambda}\dot{x}}{\hat{k}}\right| \ll 1 \quad \Rightarrow \quad \|\dot{x}\|_\infty < \frac{0.1\hat{k}}{\hat{\lambda}} \tag{4.33}$$

$$\left|\frac{\hat{\epsilon}}{\hat{k}x^{\hat{n}}}\right| \ll 1 \tag{4.34}$$

The first condition, where 0.1 is a threshold selected from experience, implies that the penetration velocity must be limited. To satisfy the second condition a reasonable minimum penetration should be selected and the identification process should be interrupted whenever the instant value of penetration becomes smaller than the chosen one.

4.6 PERSISTING EXCITATION CONDITION

The correctness of the chosen model to describe the environment behaviour influences the performance of the estimation algorithm, but the minimization of the cost function (4.8) is by itself not sufficient to guarantee that parameter estimates converge to the correct values.

The performances of recursive estimation are greatly influenced by the trajectory followed by the operated device, that should provide a sufficient level of excitation to the system. The parameters estimation requires a certain level of information to be present in the data gathered through time. This concept has been formalized in literature as *persistence of excitation*.

As described in [55], this condition requires that, independently from the past evolution of the system, the information carried by data over the next n time points spans the entire parameter space with a finite nonzero probability.

In the case of the Hunt-Crossley model, the input ψ is a nonlinear function of the position and of the velocity, so that a fairly generic motion profile is sufficient to obtain the convergence [13]. It is required nonetheless that the position changes sufficiently over time.

These conditions are certainly satisfied by the random motion of a human operator.

In regard to an automated motion, however, much simpler input signals, i.e. a sinusoidal motion profile, are also capable of producing sufficient excitation for the identification algorithm. This has been proven in [13] with randomly distributed initial conditions.

This procedure has been repeated in this work, and the results will be shown in chapter 7.

5 CONTACT EVALUATION

Initially, the experiments have been conducted with the environment in the fixed position $x_{ENV} = 0$, in order to set the focus on the performance of the estimation algorithm.
In this section we consider the more general case, in which the environment is at an unknown position $x_{ENV} \geqslant 0$. Therefore this position has to be precisely evaluated online to achieve a correct estimation of the environmental properties.

5.1 FORCE THRESHOLD DETECTION METHOD

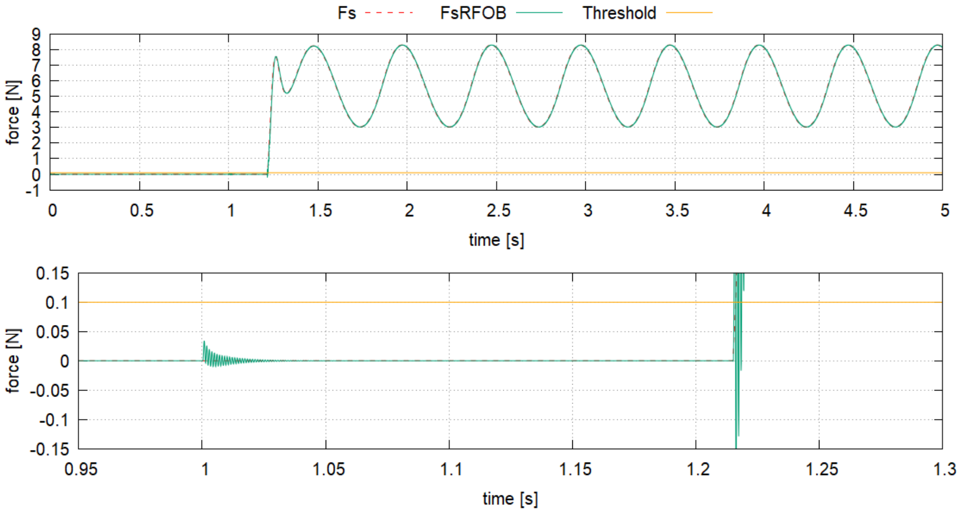


Figure 5.1: Slave force signal and threshold

One simple method to detect contact between motor and environment is based on the force response. When the slave comes into contact with the object, the force signal measured by the slave DOB/RFOB grows: a threshold can then be selected to separate contact and non-contact state.

As shown in figure 5.1, the RFOB signal (blue signal) has some disturbance, also at the beginning of the motion at $t = 1s$. Because of this, a positive value $F_{th} > 0$ must be chosen to avoid incorrect contact detection.

However this may cause some problems:

- if the threshold F_{th} (yellow signal in figure 5.1) is too low the noise on the force signal can cause false detection, sensing the environment before the actual contact takes place;
- on the other hand, a threshold value too high can result in a relevant offset between real and estimated environmental position (figure 5.2), compromising the accuracy the parameters estimation.

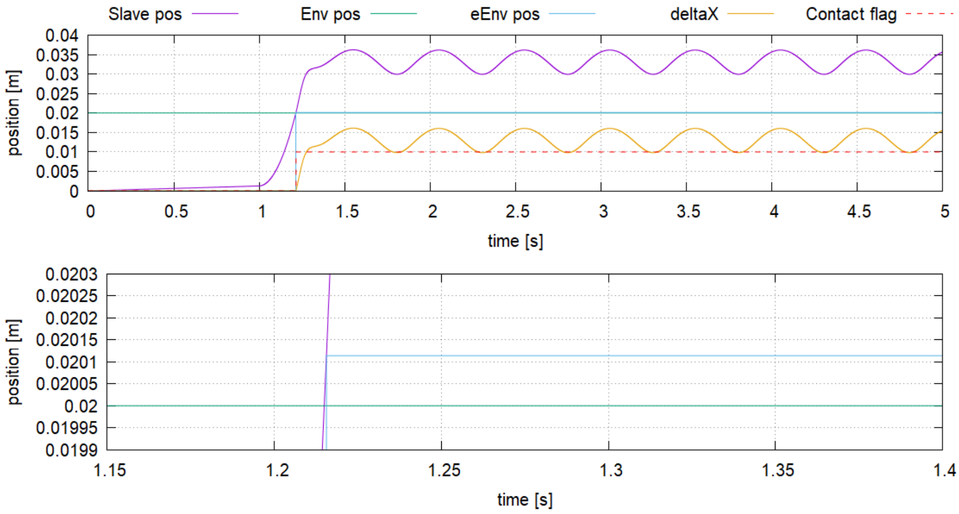


Figure 5.2: Slave position signal and estimated environment position

In figure 5.2 is represented a simulated experiment corresponding to the second case: the estimated position of the environment (light blue signal) is $\hat{x}_e \approx 0.02012\text{m}$ while the actual environmental position is set to $x_e = 0.02\text{m}$.

This phenomenon is more relevant in the experiments, as can be seen in figure 5.3 and in detail in 5.4: the force signal (purple signal) is more affected by noise, causing the contact flag (light blue signal) to change continuously state ($1 = \mathtt{true}, \quad 0 = \mathtt{false}$).

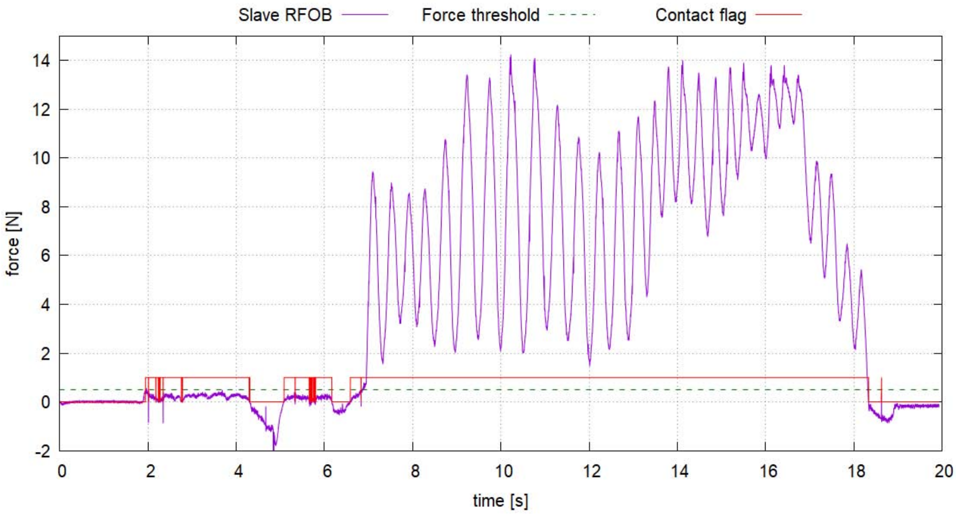


Figure 5.3: Slave force signal and threshold

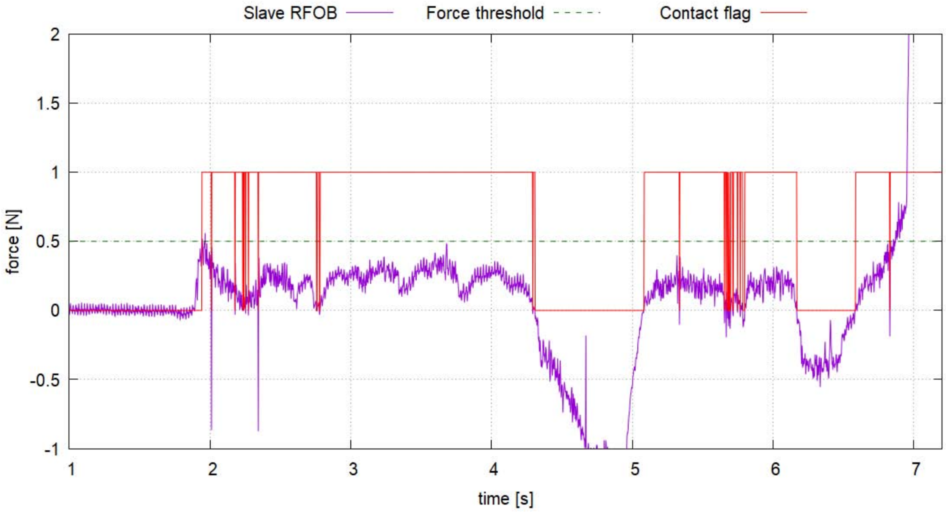


Figure 5.4: Detail of the moment of false detected contact

Thus, a higher value of force threshold must be selected. Because of this, the offset between real and estimated environmental position is higher, as shown in figure 5.5, and the estimated properties are distorted.

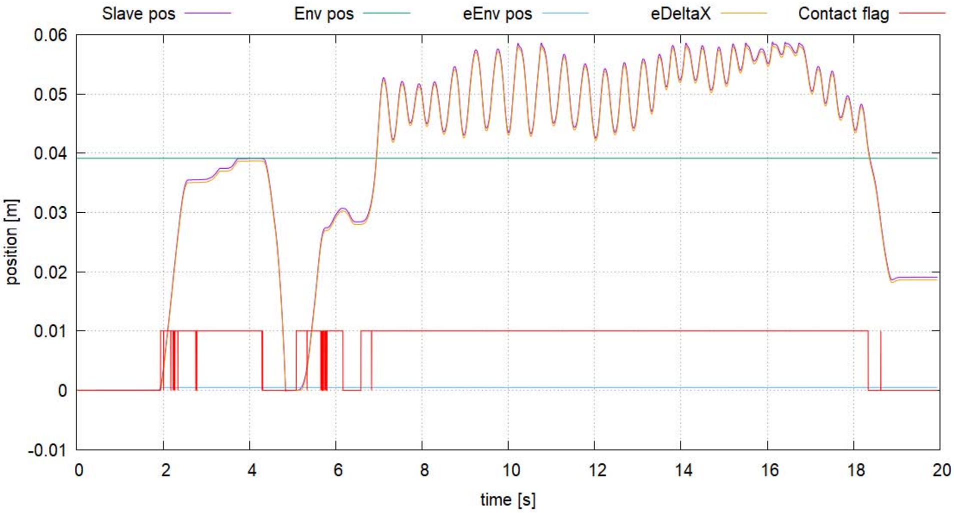


Figure 5.5: Slave position signal and estimated environment position

5.2 FFT-BASED DETECTION METHOD

In [18] a more evolved method to detect contact has been proposed. Based on the time-frequency analysis of the acceleration signal, this method relies on the damping properties of the environment.

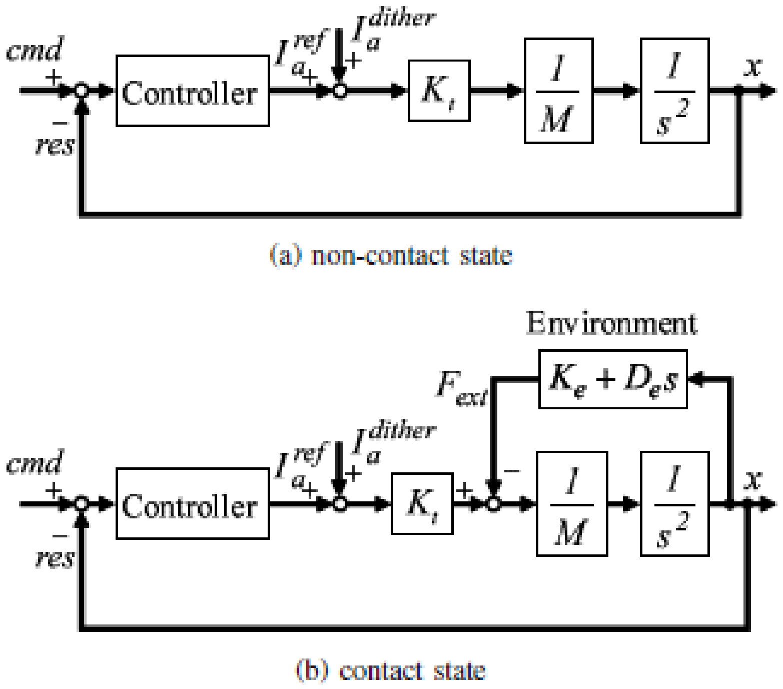
A *dither*, an intentionally applied small noise, is injected in the current signal, influencing the output signal. This small noise has been selected as a simple sine wave with a single frequency:

$$I_a^{dither} = A_d \sin(\omega_d t) \tag{5.1}$$

The effect on the output depends on both frequency and amplitude of the dither.

When the robot comes into contact with the environment this influence may be absorbed: thus monitoring the frequency components of the output, contact can be detected.

In figure 5.6 are shown two block diagrams representing the system during non-contact and contact state.



(a) non-contact state

(b) contact state

Figure 5.6: System configuration depending on contact state

The transfer function between the dither and the output is different in the two states. Considering the function from dither to acceleration, we obtain:

$$
\begin{aligned}
G_{nc}^{a}(s) &= \frac{1}{M} \\
G_{c}^{a}(s) &= \frac{s^2}{Ms^s + D_e s + K_e}
\end{aligned}
\tag{5.2}
$$

This leads to a different behaviour in the frequency domain, as shown in figure 5.7 [18].

The low frequency component of the gain decreases greatly in the contact state: contact can then be detected based on this attenuation. To select the dither frequency the following limitations must be observed:

- it must be sufficiently low in order for the attenuation to be observed;
- it must be sufficiently high not to deteriorate the time resolution of the frequency analysis.

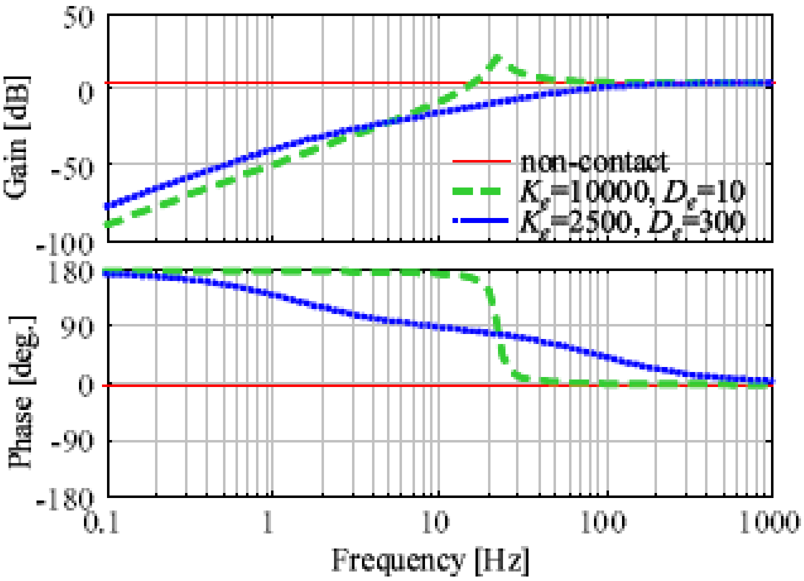


Figure 5.7: Bode diagram of the dither-acceleration transfer function

Furthermore, a difference between contact and non-contact state appears also in the phase (figure 5.7): this infomation can therefore be utilized in addiction to the gain, to improve the performance of the contact detection algorithm, especially when the limitations on the dither frequency are severe.

5.2.1 *Time-frequency analysis*

The computational cost of the FFT is not suitable for online applications. Moreover, in this case the objective of the analysis is a single frequency, rather than the complete spectrum of the output signal.

To limit the computational cost the Goertzel algorithm has been chosen to perform the time-frequency analysis in the frequency of the dither. This algorithm consists of two step, shown in the following:

$$\begin{aligned} w(n) &= 2\cos\left(\frac{2\pi m}{N}\right) w(n-1) - w(n-2) + x(n) \\ y(n) &= w(n) - e^{-j\frac{2\pi m}{N}} w(n-1) \end{aligned} \tag{5.3}$$

being

- x(n) current sample of the analysed signal;
- y(n) value of the frequency component;
- $m = \frac{2f_d}{f_s}$.

The first of the (5.3) must be implemented N times, $\forall n = 1, ..., N$, while the second is computed once after the arrival of the Nth sample.

5.2.2 *Simulation results*

The acquisition process has been analyzed in simulation, using the parameters listed in table 5.1.

Table 5.1: Simulation parameters

Sampling frequency	f_S	200 Hz
Dither frequency	f_d	10 Hz
Dither amplitude	A_{Fd}	0.1 N
Window length	N	20 samples

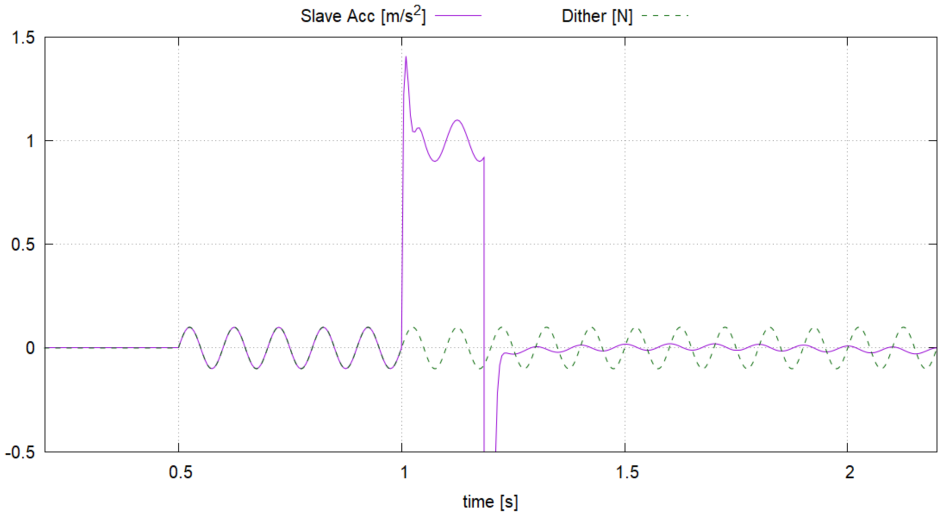


Figure 5.8: Acceleration signal and dither

In figure 5.8 is shown the acceleration signal and the dither. It is is applied from $t = 0.5s$ while the motors start moving at $t = 1s$. The influence of the dither is high until $t \approx 1.2s$, when the motor makes contact with the simulated environment.
After a brief discontinuity, caused most likely by the derivative process, the influence of the dither on the acceleration signal is reduced.

In figure 5.9 is shown the magnitude of the 10Hz component of the acceleration signal, while table 5.2 compares the values of the magnitude before and after contact is detected.

Table 5.2: Comparison of the 10Hz component magnitude in contact and non-contact state

Non-contact state	0.999626
Contact state	0.027220
Magnitude reduction	97.3%

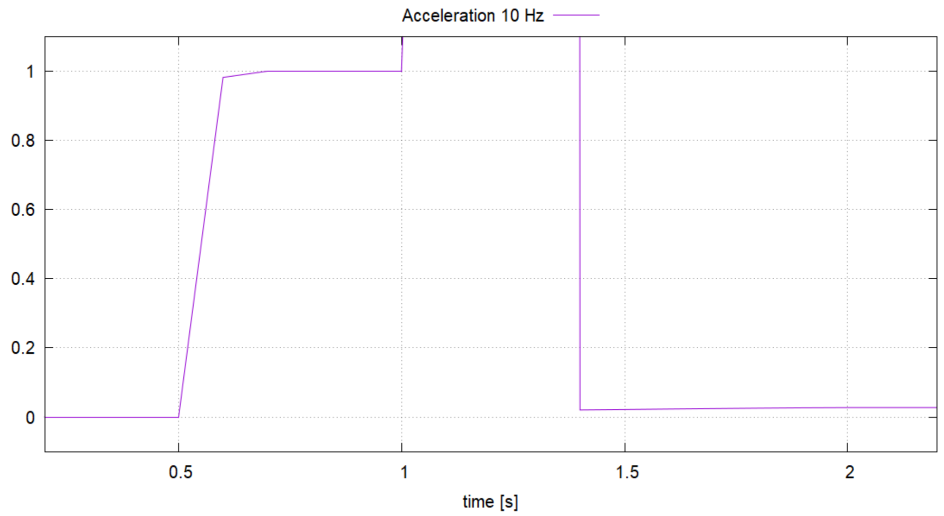


Figure 5.9: 10 Hz component of slave acceleration

In contact state the acceleration component in the dither frequency is almost completely removed, proving that this method can successfully detect contact.

6

DETECTION OF OCCURRING DEFORMATION

According to the theory of materials, an object subjected to compression experiences a loading condition that can compromise its structure. The response of the object depends highly on the mechanical properties of *stress* and *strain*.

The *stress* σ corresponds to the force applied to a material, divided by the material cross-sectional area:

$$\sigma = \frac{F}{A_0}, \quad \left[\frac{N}{m^2}\right] = [Pa] \tag{6.1}$$

On the other hand, the *strain* ϵ is defined as the deformation or displacement of a material resulting from an applied stress. It is calculated as the ratio between the variation in length caused by the applied stress and the initial length:

$$\epsilon = \frac{\Delta L}{L_0} = \frac{L - L_0}{L_0} \tag{6.2}$$

where

- L_0 is the original length of the material ($[mm]$);
- L is the material length after the load is applied ($[mm]$).

In figure 6.1 is shown a stress-strain diagram describing the behaviour of an object under loading condition. The left part of the diagram represents the elastic region: the applied stress causes a deformation in the material, but when the force is removed the object can return to its original shape without permanent deformation.

Once the stress grows past the value corresponding to the yield

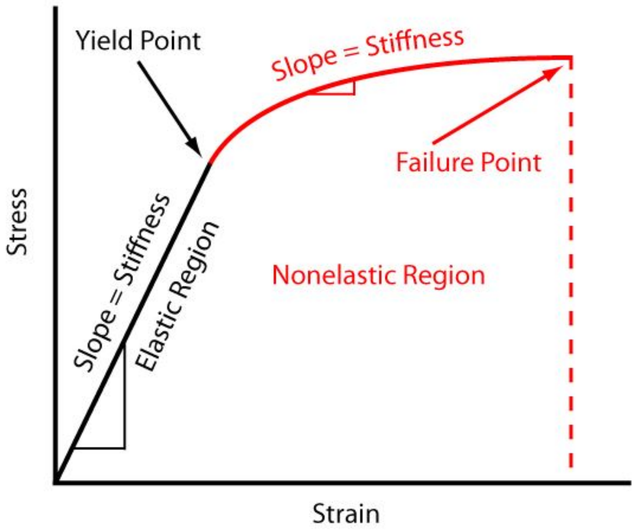


Figure 6.1: Stress-strain diagram

point the object enters the plastic region: a percentage of the deformation becomes permanent, and persists even if the force is removed.
If the stress grows even further the material will eventually reach the failure point, in which the rupture occurs.

As can be noticed in picture 6.1 the slope of the curve corresponds to the stiffness of the material. This stress-strain relationship is known as *Hooke's Law*, and in this region, the slope is referred to as the *modulus of elasticity*, or *Young's modulus*, denoted E.

$$E = \frac{\sigma}{\epsilon} \tag{6.3}$$

While transitioning from the elastic region to the plastic region the slope decreases, reaching zero at the failure point.
Considering this aspect, the environment impedance estimation method described in chapter 4 can be utilized to detect this transition.

To focus on this aspect some experiments have been conducted: the linear motors controlled via Bilateral Control described in the previous chapters were manipulated by the operator to impact and deform the environment made of clay: this is highly ductile and can be deformed easily.

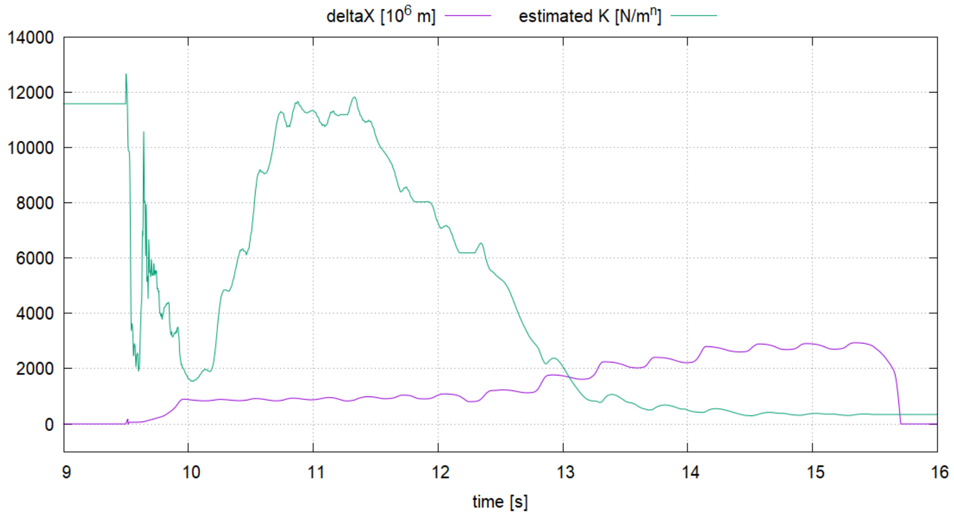


Figure 6.2: Penetration depth and estimated stiffness

In figure 6.2 are compared the signals corresponding to the estimated value of stiffness and the penetration depth, magnified with a factor of 10^6 to allow the comparison.

The contact between motor and environment occurs approximately at $t = 9.5s$, and the estimation procedure initiates. After a brief transition the estimated stiffness converges to a value between 11000 and $12000N/m$. At $t \approx 11.5s$ the motor is penetrating into the clay, causing a plastic deformation, as the value of the estimated stiffness decays rapidly. In table 6.1 are compared the values estimated before and after the deformation.

An occurring deformation of the environment greatly affects the stiffness of the environment, and the presented estimation procedure can detect this variation: using this method an oc-

Table 6.1: Stiffness before and after deformation

Before deformation	11196.692148 N/m
After deformation	380.078845 N/m
Magnitude reduction	**96.6%**

curring plastic deformation can be detected while performing other operations on the environment, allowing the control system to enable a chosen countermeasure.

7 EXPERIMENTAL RESULTS

In this section the results from simulations and experiments are compared.

7.1 EXPERIMENTAL SETUP

The system used in this thesis is shown in 7.1: it consists of two identical 1-DOF linear motors connected by bilateral control.

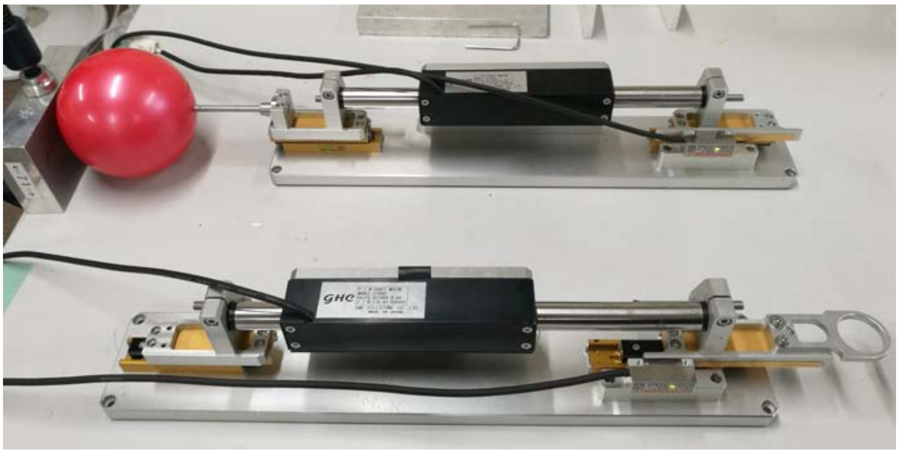


Figure 7.1: System used

Figure 7.2 is a schematic representation of the complete system. The actor operates the master system **M**, that communicates with the PC: the master position x_M and the force F_M given by the actor to the master are processed by the control system and reproduced on the slave **S** as $x_S = x_M^*$, $F_S = F_M^*$. The superscript * in figure 7.2 is given to the components reproduced by the control system.

The slave system will move accordingly to the motion given by the operator to the master. When the condition $x_S = x_E$ is true, it comes into contact with the environment **E**: the slave exerts on the object a force $F_S = F_M$, while, for the law of action and reaction, the environment responds with a force $F_E = -F_S$. This force is reproduced by the control system on the master as F_E^*, so that the actor experiences the reaction of the environment as if he was directly in contact with it.

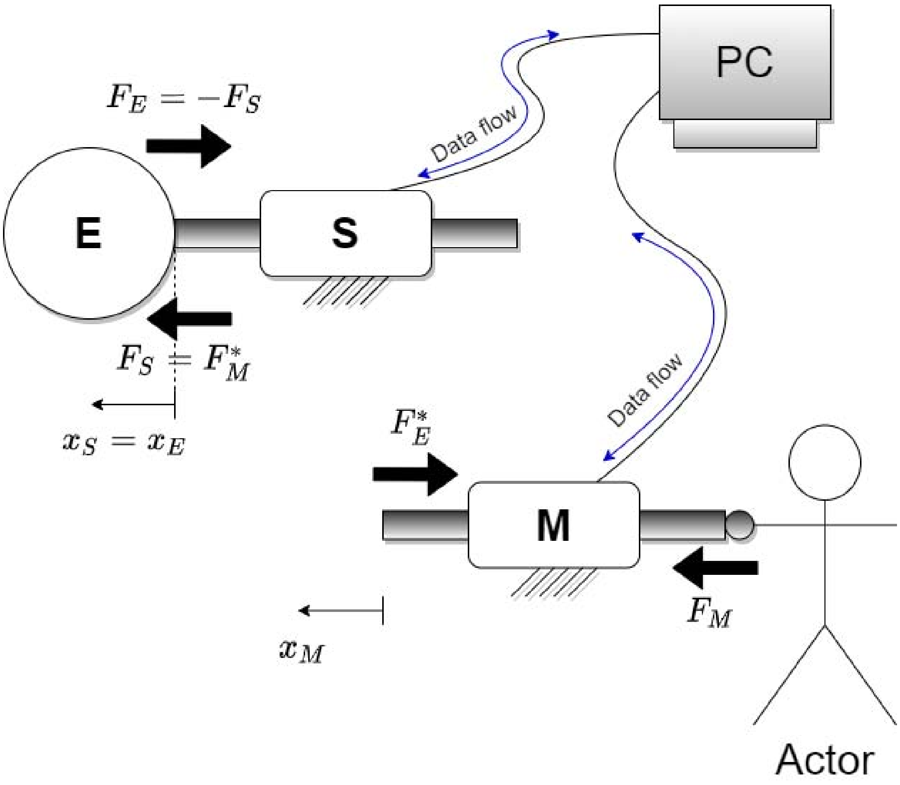


Figure 7.2: System scheme

The characteristics of the motors are given in 7.1, while the specifics of the encoders are given in table 7.2.

The control system utilized is bilateral control based on acceleration dimension [5]. This allows to couple the two motors in a master-slave relationship: the operator moves one system (master), and the same trajectory is mimicked automatically on the other system (slave). The contact with the environment is performed only by the latter.

Table 7.1: Linear motor characteristics

Model	GHC S160Q
Rated voltage	240 V
Rated current	0.62 A
Rated force	20 N
Force constant	33 N/A
Shaft diameter	16 ± 0.1 mm

Table 7.2: Encoder characteristics

Model	Renishaw RGH24Y
Power supply	5 V ± 5%
Output frequency	12MHz µm
Resolution	0.1 µm

7.2 IDENTIFICATION OF DISTURBANCES

To decouple the DOB and RFOB signals the constant value of the disturbances described in section 2 is detected at the beginning of the experiment.
The motors are in a standstill condition, with no contact: in this condition either the force from the operator and the reaction force from the environment are zero, so the DOB signal represents only the disturbances.

The average of this signal is calculated to ignore the 50Hz oscillation and isolate the constant component. This is then subtracted to the RFOB signal to isolate the external force, as shown in figure 7.3.

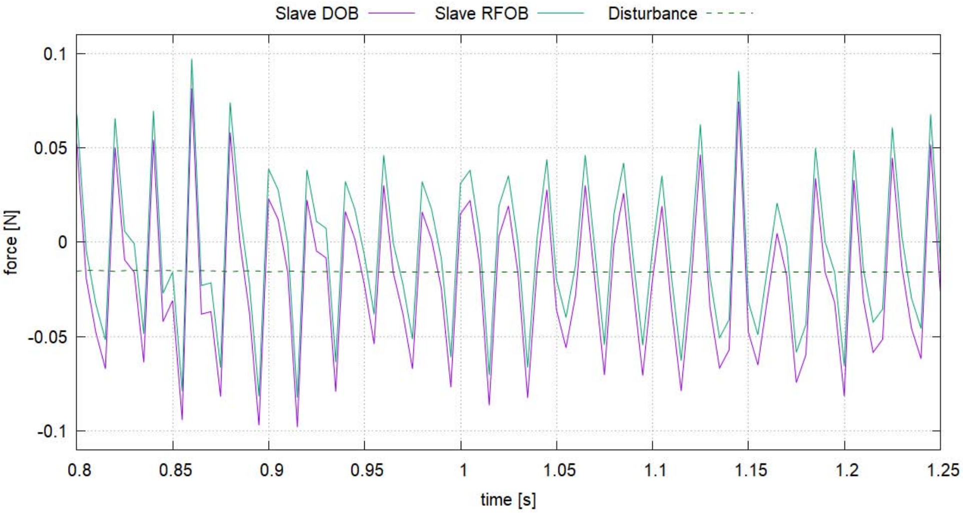


Figure 7.3: Disturbance detection

The DOB force signal (in purple) presents a clear oscillating behaviour, with fundamental frequency $f_I = 50\text{Hz}$: this is due to the oscillating current deriving from the power source. The average of this signal (in red) is calculated online up to the time $t = 1$, then it is maintained constant and utilized in the RFOB structure to, so that the RFOB signal will only contain the information on the external forces (in green).

7.3 BILATERAL CONTROL

Then the correct behaviour of the control structure is confirmed. The force signals from master and slave during simulation are shown in figure 7.4 and in detail in figure 7.5, while the position signals are shown in figure 7.6.

The DOB/RFOB structures are able to quickly estimate the external forces applied to the motors by the operator and the environment, as well as the constant disturbances applied to mimic the real system. In figure 7.4 the signals from the master and slave RFOB structures (in green and orange) are compared with the simulated force signals for master and slave, F_M (in purple) and F_S (in light blue). The component F_S is simulated using the environmental model described in (3.8) (chapter 3). The value of the coefficients are reported in table 7.3.

Table 7.3: Value of the coefficients of the simulated environment

Stiffness coefficient	k	$10^3 N/m^n$
Damping coefficient	λ	$10^4 Ns/m^{n+1}$
Exponent	n	1.2

The estimated force signals overlay perfectly with the true force signals, guaranteeing a fast and correct estimation.

The error between the master force and the slave force is shown in gray. When the applied force is constant, the error reaches zero after a short transient, whereas when the input signal becomes sinusoidal a small error persists. This is probably due to the behaviour of the force controller inside the bilateral control structure: the values of the force gain are limited, and it is usually set to $K_F = 1$ for stability reasons. This, however, will not affect the environment properties estimation, because the procedure will utilize the force response from the object, that is coherent with the actual position of the motors, thus with the penetration depth.

In figure 7.5 the initial transient is shown, focusing on the effect of an inserted disturbance both on the master and on the slave. As can be seen, the DOB structures estimate the sum of the applied force and the disturbance. For example, on the master system are applied a constant force $F_M = 10N$ (in purple)

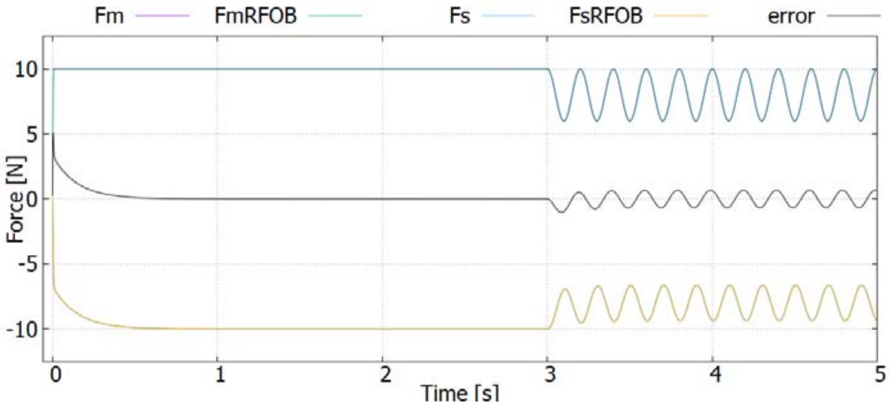


Figure 7.4: Force signals for bilateral control

and a constant disturbance $F_M^{dist} = 2N$ (in green). The master DOB estimates (red signal):

$$F_M^{DOB} = 12N = F_M + F_M^{dist} \tag{7.1}$$

while the master RFOB isolates only the external force (blue signal):

$$F_M^{RFOB} = 10N = F_M = F_M^{DOB} - F_M^{dist} \tag{7.2}$$

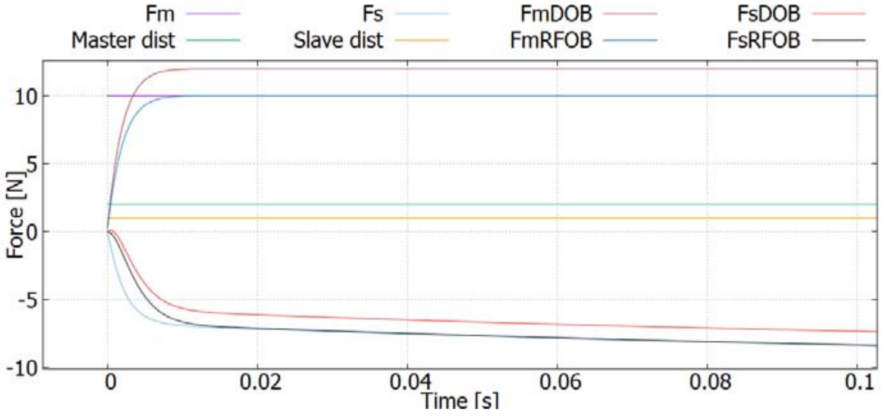


Figure 7.5: Initial transition phase

Furthermore, the bilateral control is correctly executed: as can be seen in figure 7.6, the position of the two motors overlaps after a short transition phase, and the difference between the force signals is sufficiently close to zero even when a sinusoidal force command is applied to the master.

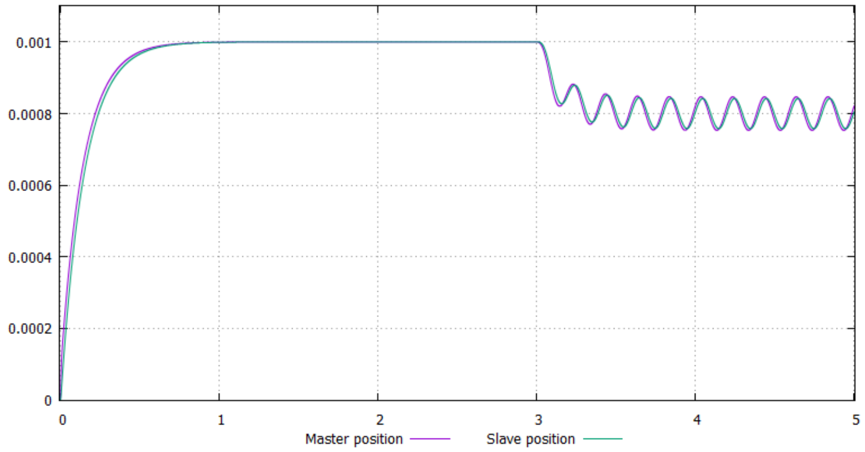


Figure 7.6: Position signals for bilateral control

The correspondent experimental results are shown in 7.7 and 7.8.

In detail, 7.7 shows the RFOB force signal for master and slave, while in figure 7.8 are shown the overlapping position signals of master and slave.

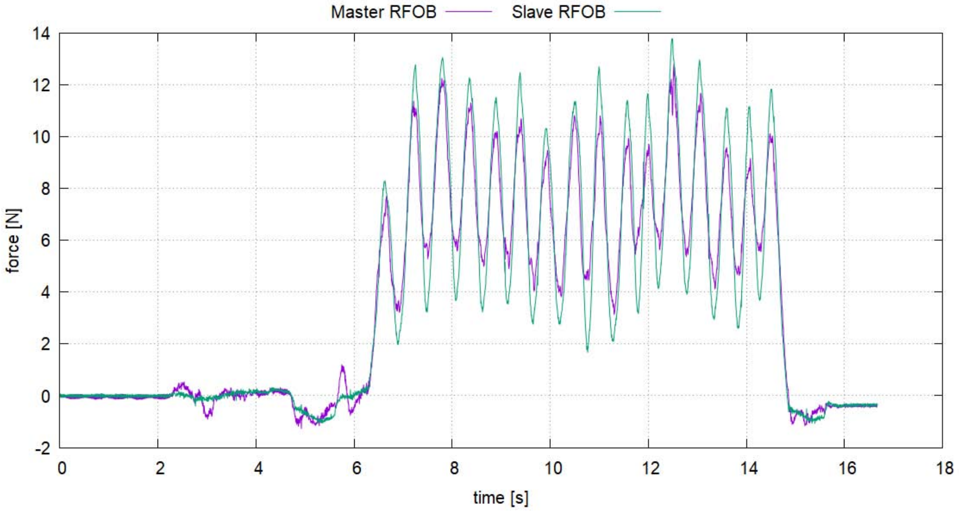


Figure 7.7: Force signals for bilateral control

As can be seen there is still a difference between the master force (in purple) and the slave force (in green), as seen in the simulations. However, this error is negligible, as the master motion is correctly reproduced by the slave: as shown in figure 7.8 the master and slave position signals overlay perfectly.

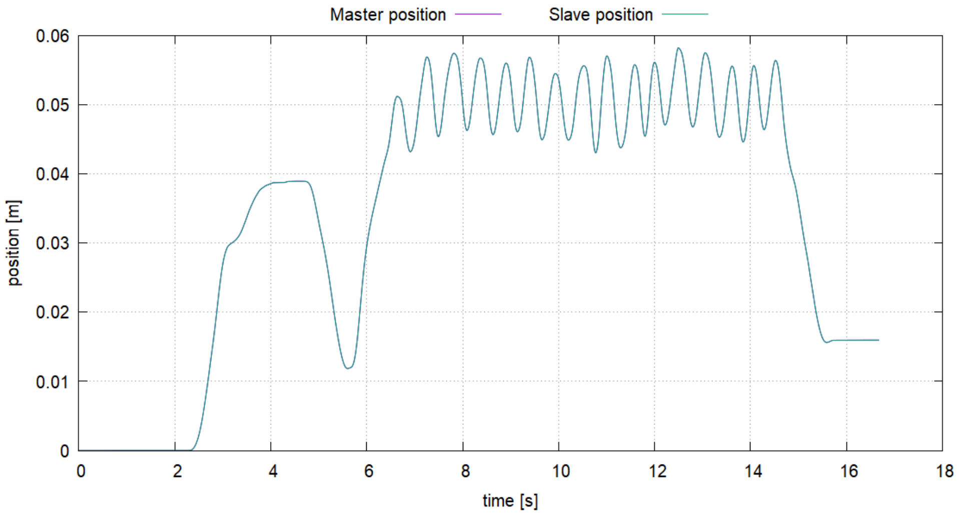


Figure 7.8: Position signals for bilateral control

Therefore, the behaviour of bilateral control structure is satisfactory and we can proceed with the environment analysis.

7.4 ENVIRONMENT ANALYSIS

The environment position is set as $x_{ENV} = 0$, so that the estimation process can start as soon as the motion begins.
Initially the force command is set to $F_M = 0N$ and the disturbances are estimated, in order to initialize the RFOB structure. Then the estimation process is initiated: to maximize the efficiency of the algorithm a sinusoidal force command has been used, for the reasons described in section 4.6.

During the simulations a range of different properties has been used for the virtual environment, to test the impact of the actual stiffness and viscosity on the estimation algorithm.
During the experiments various objects with different characteristics have been used, in detail:

- a sponge
- a rubber ball
- a 1kg med ball
- some clay.

In this thesis are shown the results only for the rubber ball, considered the most simple and thus straightforward case for the current purposes.

7.4.1 *Single-stage RLS estimation*

At first the Kelvin-Voigt environmental model and the single-stage RLS method have been used.
The simulation has been repeated with different values of the environment parameters. Here are presented the results with parameters set to $K = 10^3, D = 10^2$.

Figure 7.9 shows the behaviour in position, with the master and slave position signals overlaying. Figure 7.10 shows the estimated values of stiffness and viscosity constants $\hat{K}, \hat{D}$ (in light blue and yellow, respectively) together with the real values K and D (in purple and green): the estimated parameters converge rapidly to the correct values.
In figure 7.11 is shown the behaviour of the forgetting factor β: at the beginning of the estimation, when the error e is large, the forgetting factor decreases from 1 to approximately 0.998, increasing the speed of the estimation. Then, when convergence is reached, it settles to 1.

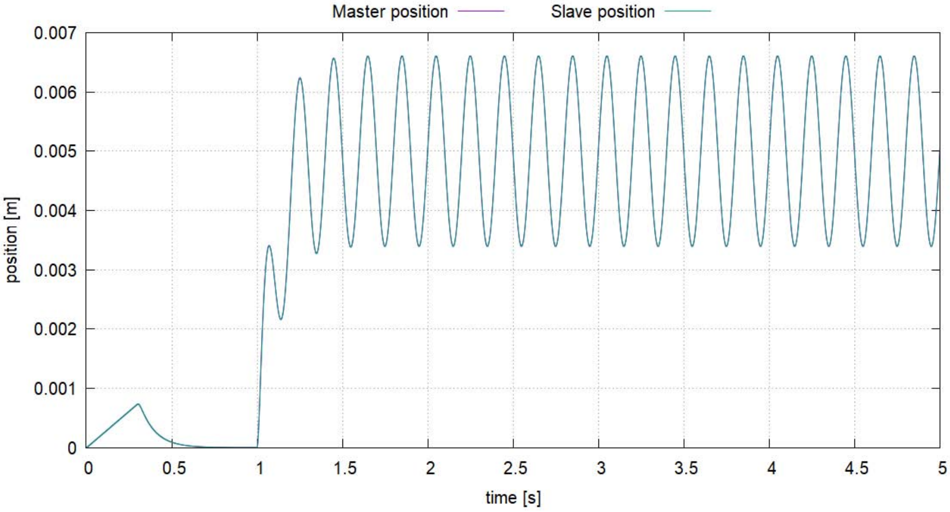


Figure 7.9: Position signal

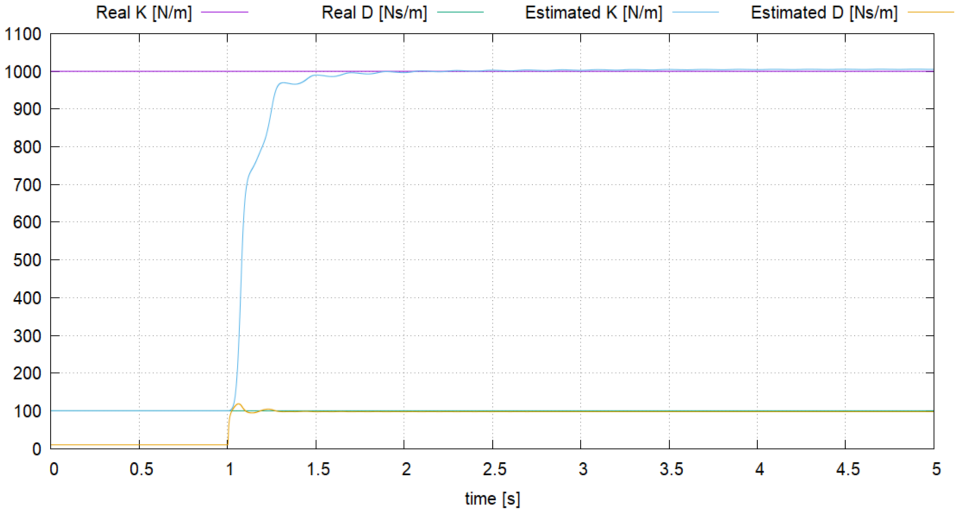


Figure 7.10: Estimated environment characteristics

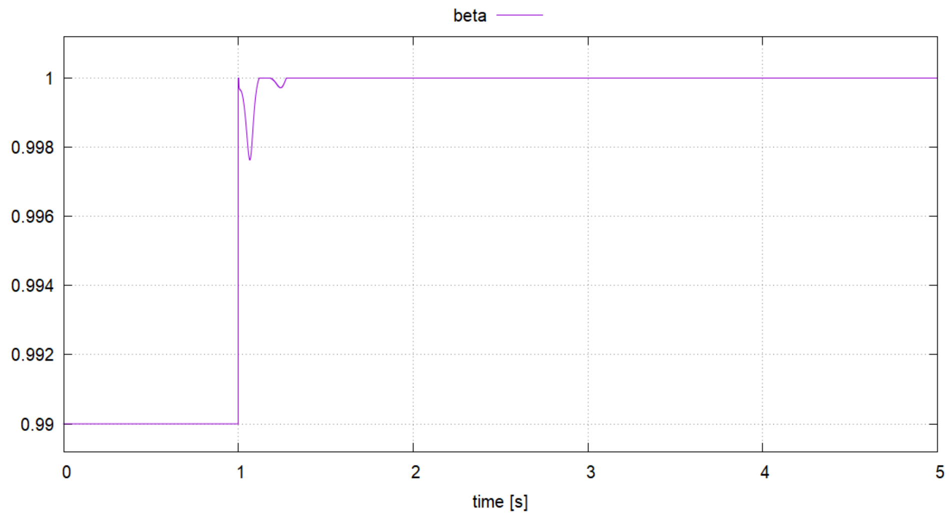


Figure 7.11: Variation of forgetting factor β

The algorithm is able to precisely estimate the environment characteristics. The estimated force is compared with the signal from the slave RFOB in 7.12: after a short transient convergence is reached, with an estimation error lower than 1N.

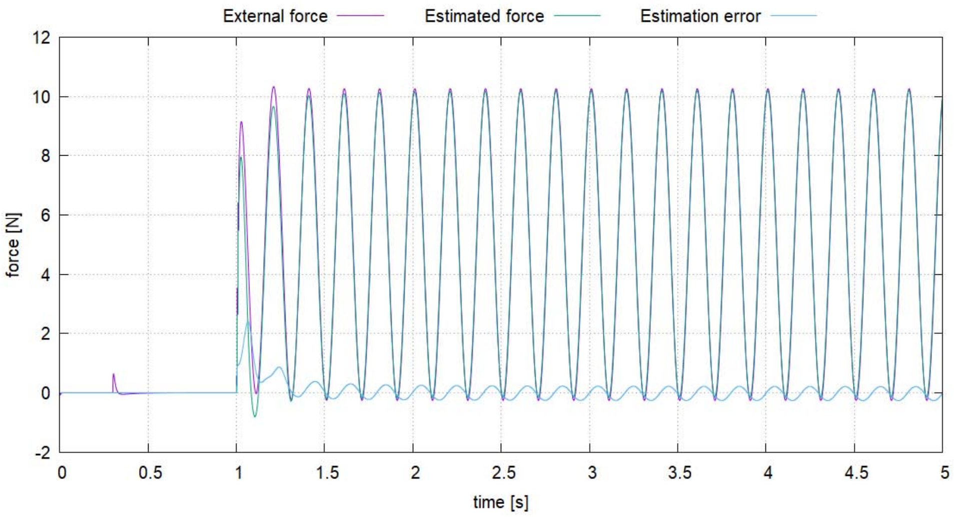


Figure 7.12: Estimated force

The results from the experiments are shown in figure 7.13 and 7.14.

Figure 7.13 shows the estimated characteristics.The experiment has been repeated several times, to confirm convergence

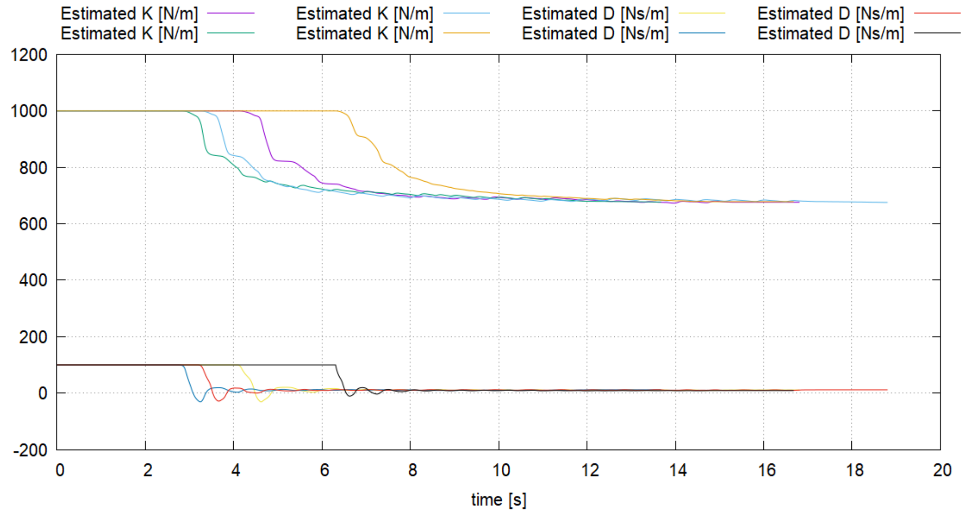


Figure 7.13: Estimated environment characteristics

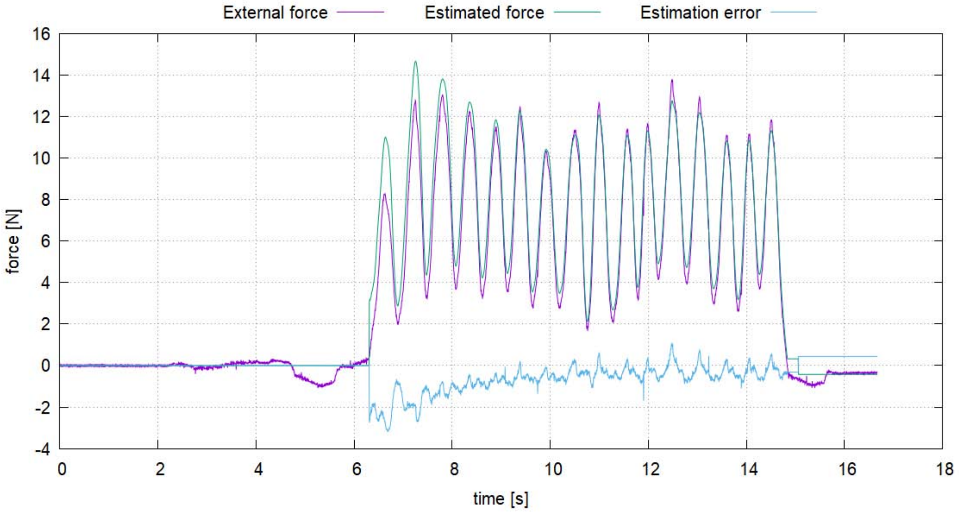


Figure 7.14: Estimated force

to the correct values.

Figure 7.14 shows the slave RFOB signal (in purple) confronted with the estimated force (in green), and the relative estimation error (light blue).

With an estimation error below 2N at convergence, the estimation process is considered successful and the study can proceed to the next step.

7.4.2 *Double-stage RLS estimation*

The experiments have then been repeated using the Hunt-Crossley environmental model and the double-stage RLS algorithm.

At first, only the estimator Γ_1 has been evaluated, in order to detect any discrepancy in its behavior.

Figure 7.15 shows the estimated environmental characteristics $\hat{k}, \hat{\lambda}$, while the estimated exponent $\hat{n}$ has been set to match the real exponent n. Here are presented the results with parameters set to $k = 10^4, \lambda = 10^3, n = 1.2$.

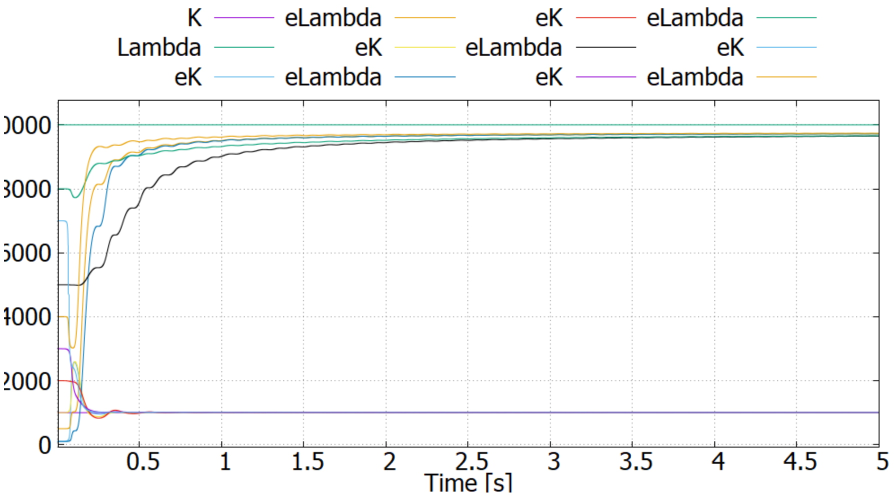


Figure 7.15: Estimated environment characteristics k, λ

The algorithm successfully estimated the parameters, independently from the initial values k_0, λ_0.

Then the second estimator Γ_2 has been connected, in order to estimate all three parameters. The simulation results are shown in figure 7.16 and figure 7.17.

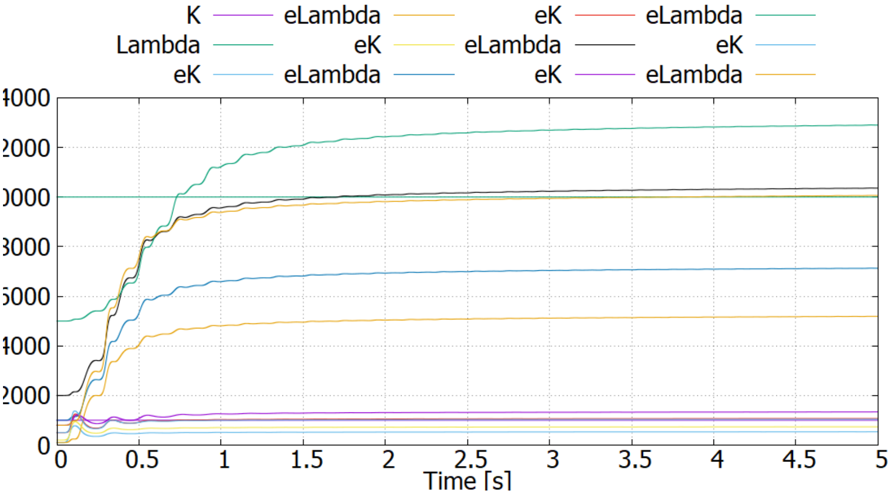


Figure 7.16: Estimated environment characteristics k, λ

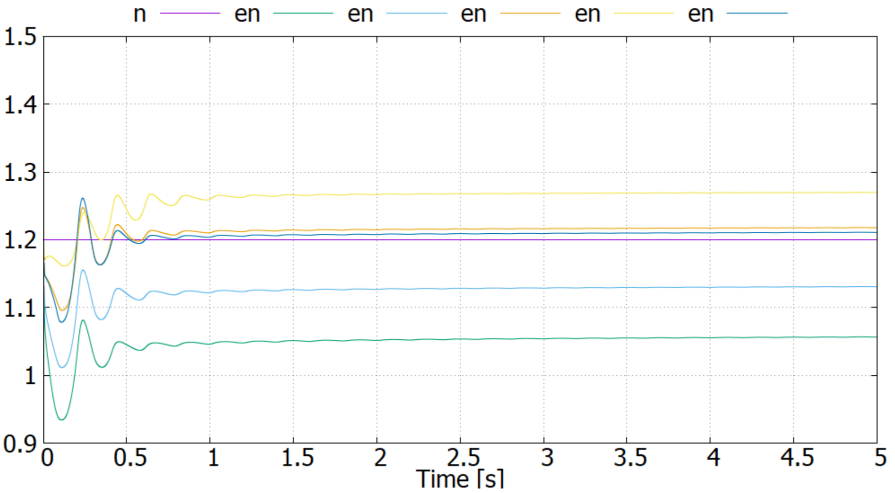


Figure 7.17: Estimated exponent n and forgetting factor β

In this case the choice of the initial values k_0, λ_0 greatly affects the values at which the algorithm converges.
As shown in 7.18, however, the estimated force corresponds to the force from the environment.

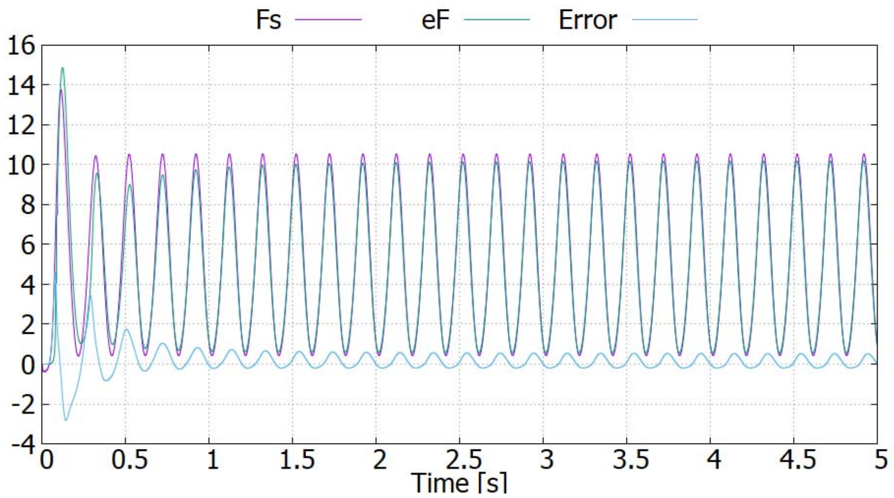


Figure 7.18: Estimated force

For the estimation to be unbiased the conditions presented in section 4.5.2 must be satisfied. Figure 7.19 shows the two conditions in a correct case. The two sides of equation (4.28) are approximately equal to each other during the whole execution. The condition in equation (4.29) is not met at the beginning of the estimation process, but after a short period of time $x^{\delta n}$ converges close to 1.

Instead in figure 7.20 is shown a case in which the two conditions are not met. The left and right side of equation (4.28) are very different, and $x^{\delta n}$ converges to a value between 1 and 2. Because of this, the approximations utilized to derive the estimation algorithm lose their validity, and the algorithm converges to incorrect values.

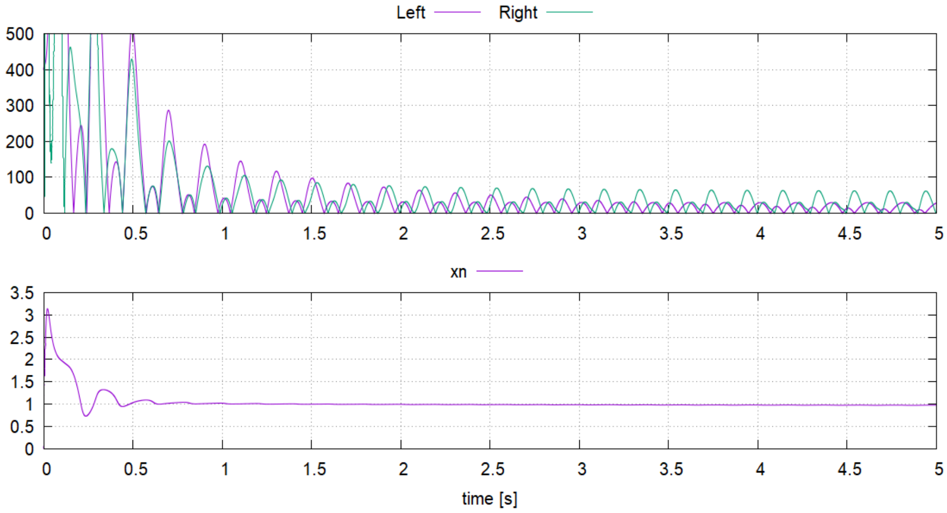


Figure 7.19: Conditions (4.28) and (4.29) in a correct case

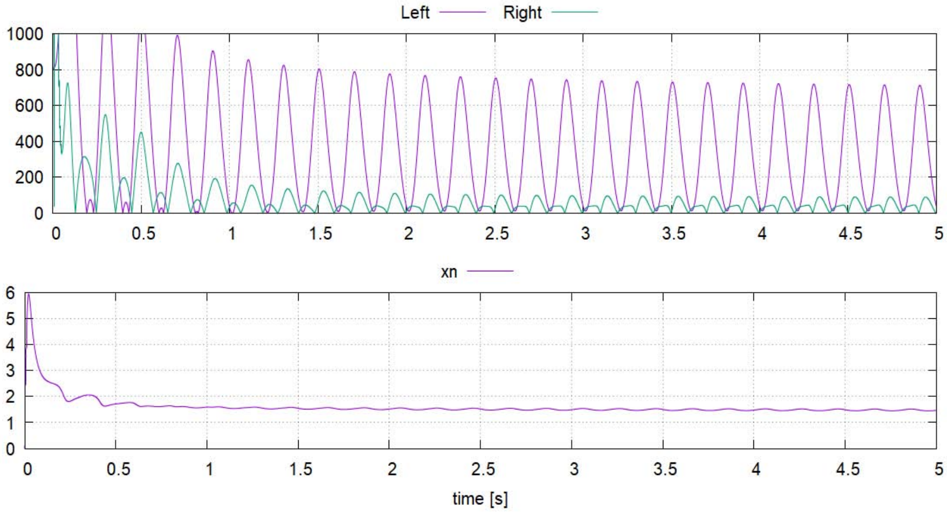


Figure 7.20: Conditions (4.28) and (4.29) in an incorrect case

The experimental results are shown from figure 7.21 to figure 7.24.

In figures 7.21, 7.22 and 7.23 are presented the estimated environmental properties: as expected from the simulations, the estimator converges to different values depending on the initial condition k_0, λ_0.

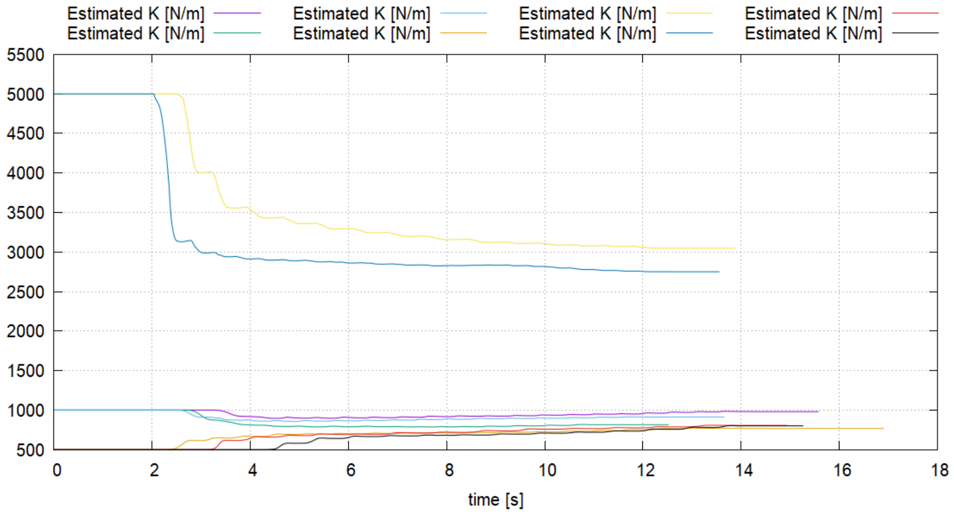


Figure 7.21: Estimated environment characteristic k

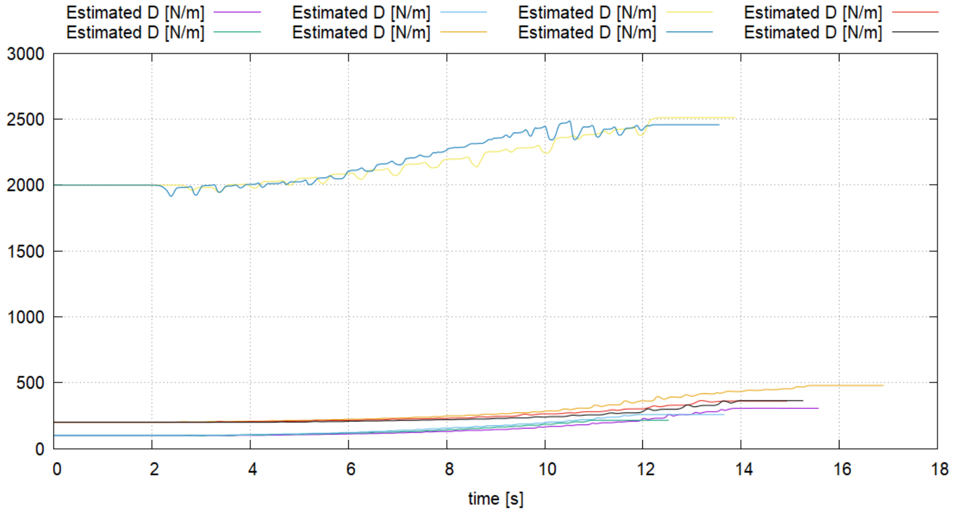


Figure 7.22: Estimated environment characteristics λ

The estimated force is coherent with the force signal form RFOB in most cases, with estimation error ranging from 0.5N

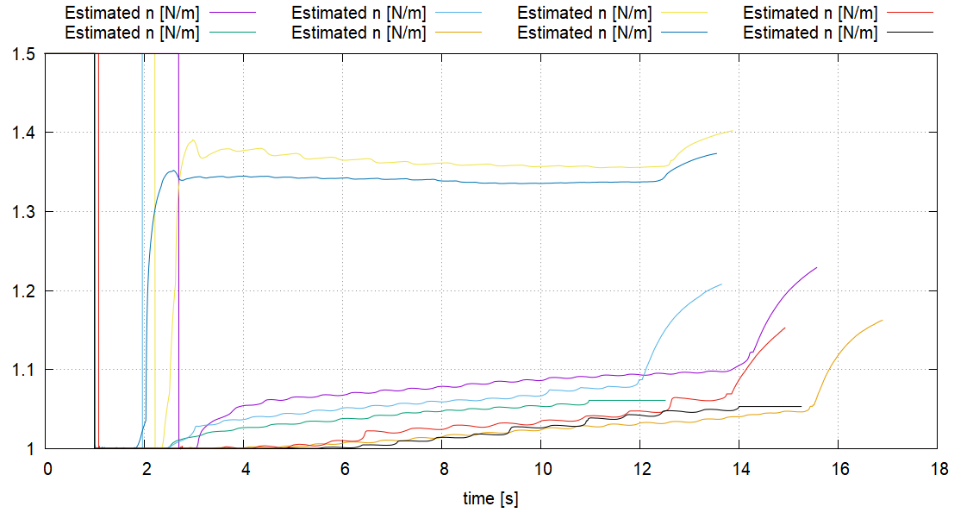


Figure 7.23: Estimated exponent n

to 5N. In figure 7.24 is shown one of the tests with the lower error.

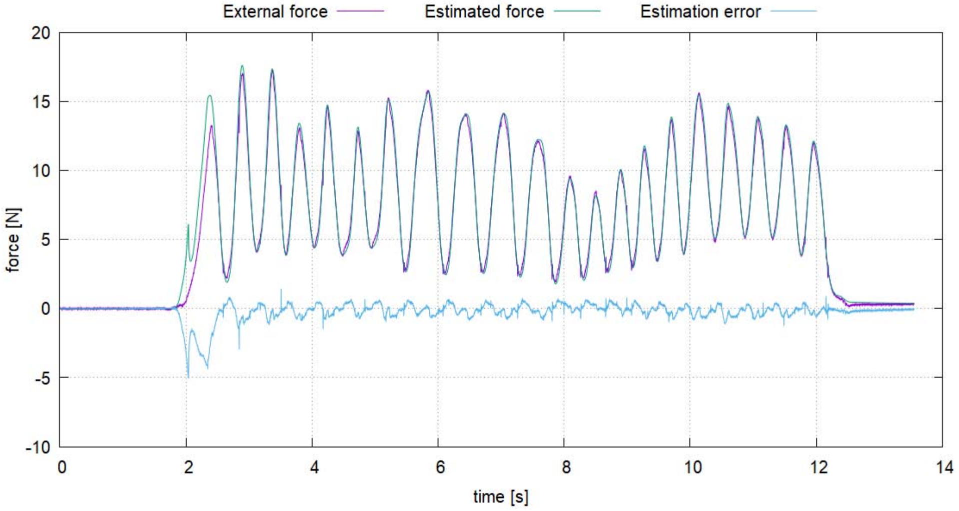


Figure 7.24: Estimated force

As expected, the estimation results are inconsistent from trial to trial, proving that this method is not suitable for the purpose of this thesis.

7.4.3 *Single-stage logarithmic estimation*

The problems derived from the feedback interconnection of the two estimators can be avoided using the logarithmic approach. Figure 7.25 and 7.26 show the comparison between the estimated parameters and the real values. Convergence is achieved regardless of the initial condition k_0, λ_0, n_0.

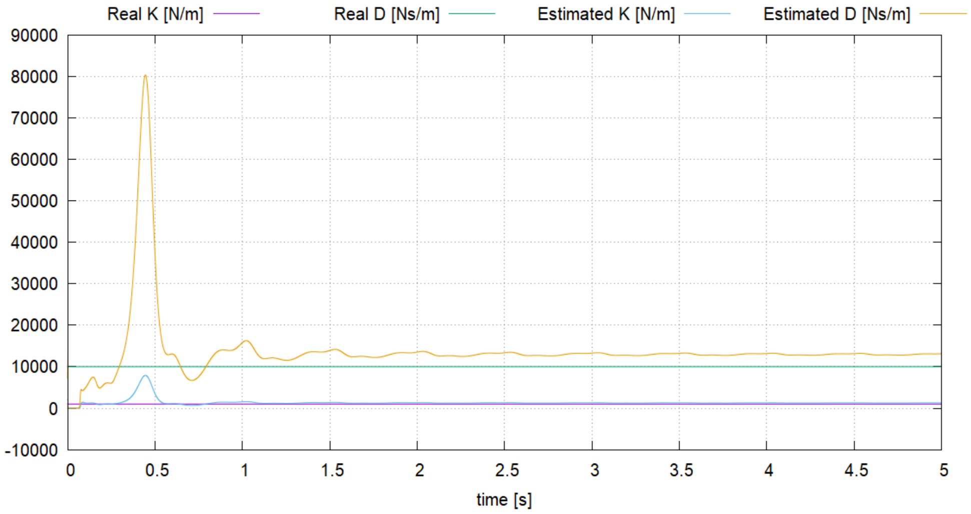


Figure 7.25: Estimated environment characteristics k, λ

As discussed in section 4.5.3, a minimum penetration depth is maintained, as can be seen in figure 7.27, in order to avoid singularities.

The condition expressed by equation (4.33) is verified, except for a brief moment at the beginning of the estimation, as shown in figure 7.28.

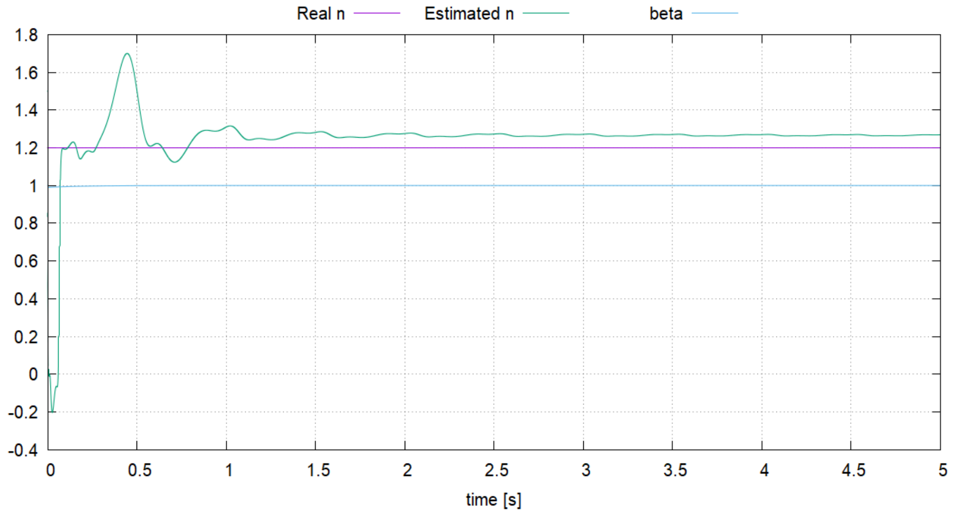


Figure 7.26: Estimated exponent n and forgetting factor β

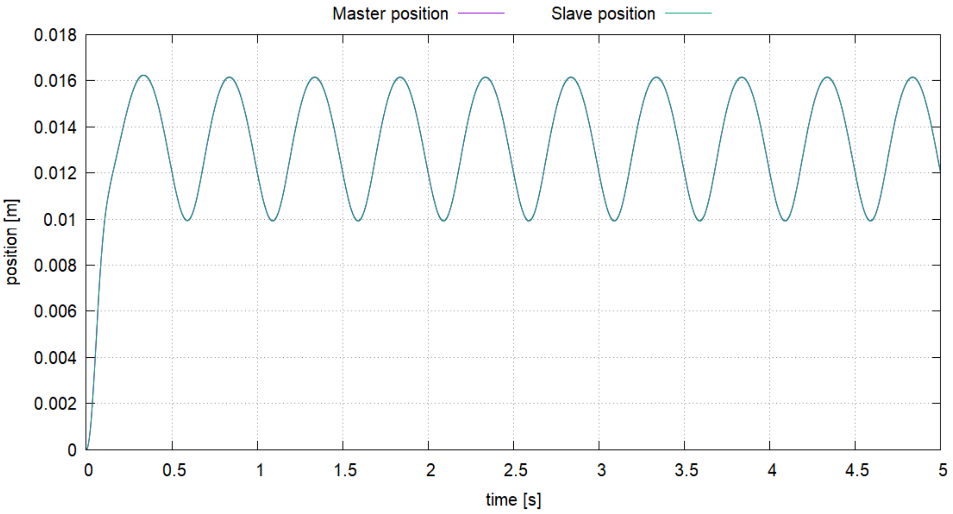


Figure 7.27: Position during estimation

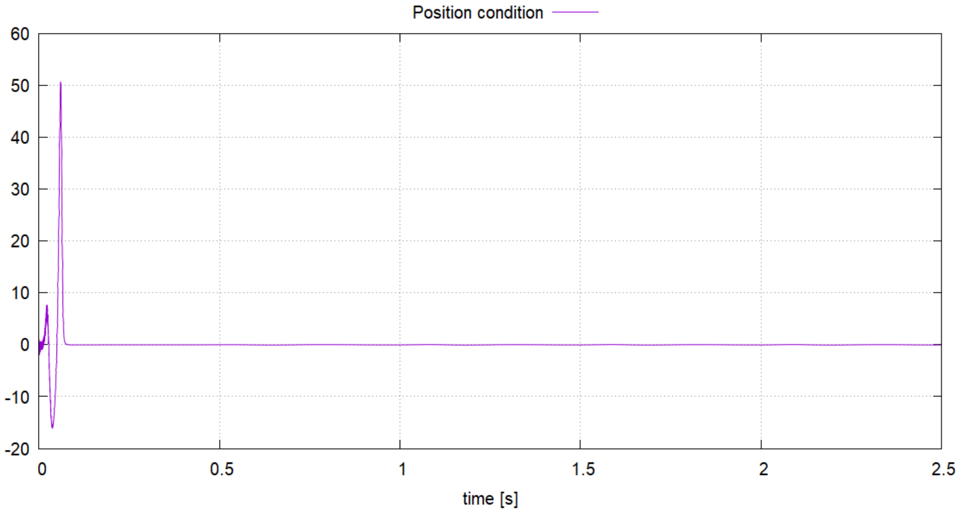


Figure 7.28: Equation (4.33) during estimation

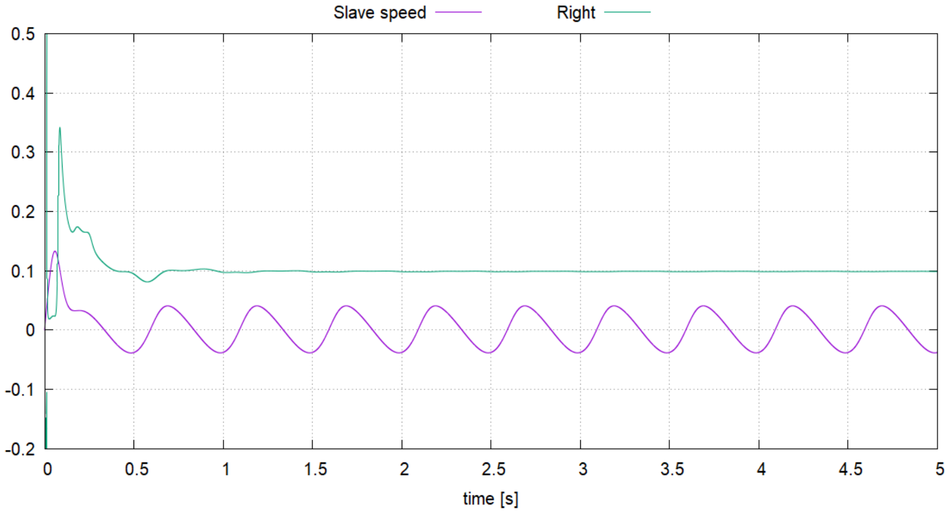


Figure 7.29: Equation (4.34) during estimation

Experimental results are shown in figure 7.30, 7.31 and 7.32.

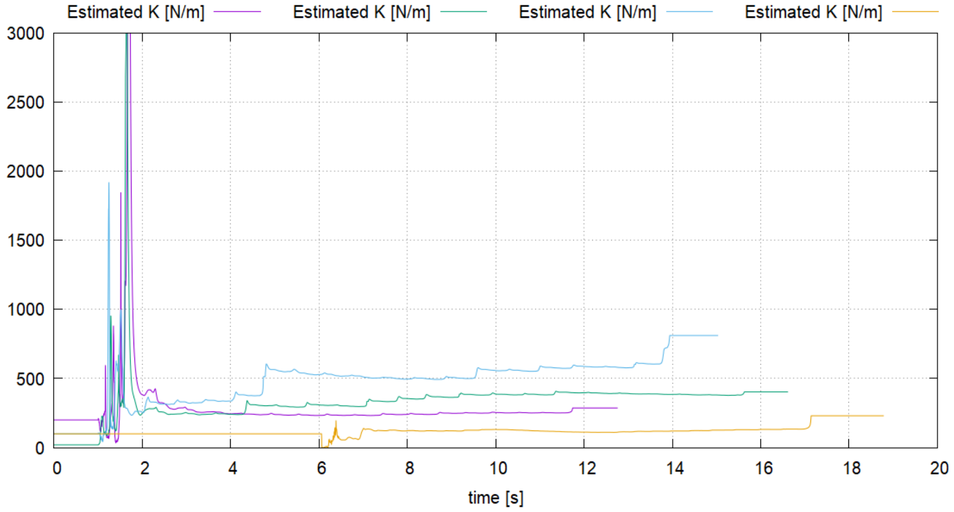


Figure 7.30: Estimated environment characteristic k

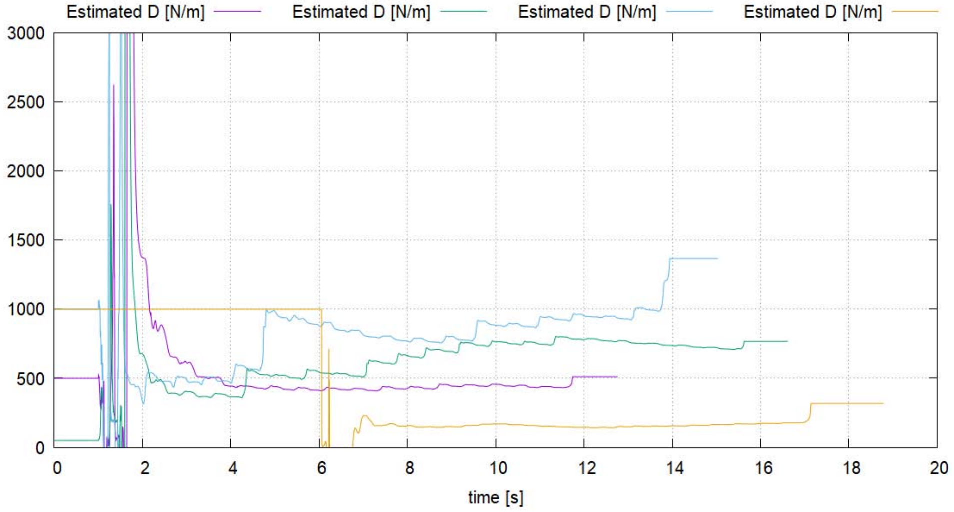


Figure 7.31: Estimated environment characteristics λ

Even though the results are not as good as in the simulations, the estimation does converge in a limited range of values, much more than utilizing the previous method.

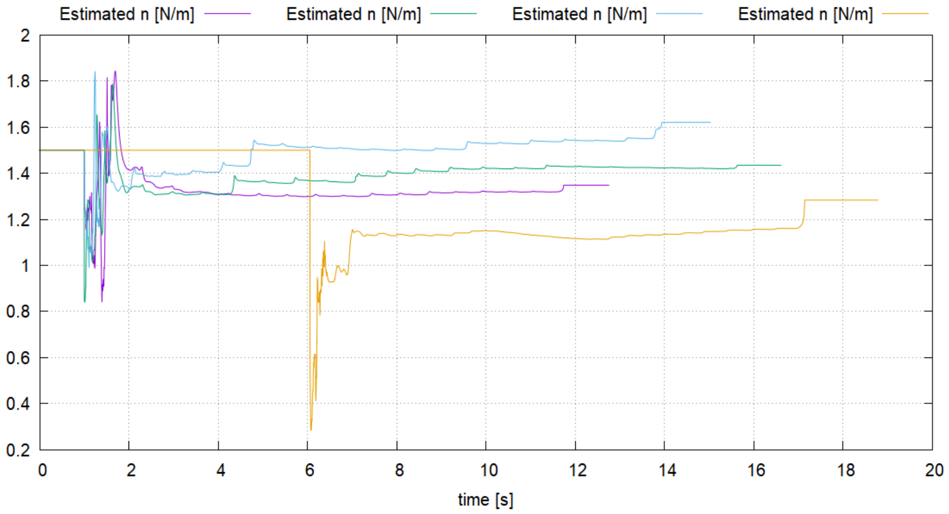


Figure 7.32: Estimated exponent n

CONCLUSION

In this thesis bilateral control based on acceleration dimension has successfully been implemented.

To analyze the properties of an unknown environment, two different models have been compared: while the Kelvin-Voigt model is beneficial thanks to its simplicity, the Hunt-Crossley model has been proven to be more accurate, especially when describing soft objects.

Then, three estimation methods based on the Recursive Least Square method have been investigated. During the identification of the Kelvin-Voigt constants no major issues have been encountered, while recognition of the Hunt-Crossley properties has been more challenging: the double-stage algorithm showed some limits, however the logarithmic approach has been proven to be a suitable method for this task.

Then identification of contact between motor and environment has been investigated: the method based on a force threshold resulted ineffective for the purposes of this thesis, but the method based on the time-frequency analysis has proven to be reliable.

At last, the estimation algorithm was proven able to identify an occurring plastic deformation of the environment under study, allowing for the deployment of a suitable countermeasure.

APPENDIX

A

CODE FOR SIMULATION

```c
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <stdbool.h>

// Initialize system data

// System data
#define Mn 0.5              // mass of the motor [kG]
#define Kt 33.0             // torque constant [N/A]
#define T 0.0001            // sampling period [s]
#define STEP 100000         // Number of timesteps

// Output files
#define WRITEFILE "output.dat"              // Name of
    output file for position and force
#define WRITEFILEBF "bilateF.dat"           // Name of
    output file for bilateral control
#define WRITEFILEBP "bilateP.dat"           // Name of
    output file for bilateral control
#define WRITEFILE2 "environment.dat"        // Name of
    output file for environment estimation

// Force command and disturbances
#define F_CMD 10.0              // Force reference [N]
#define FMDIS 2                 // Master external disturb
```

```
#define FSDIS 1                 // Slave external disturb

// Switches for environmental model
#define ENVIRONMENT 1           // 0 = KV, 1 = HC
#define DIFF 1                  // 0 = instantaneous, 1 =
    differential

// Control gains and cutoff frequencies
#define Kf 1                    // Force Gain
#define Kp 5000                 // Position Gain
#define Kv 200                  // Speed Gain
#define Ka 0                    // Acceleration Gain
#define Gobs 600                // DOB filter bandwidth
#define Greac 2000              // RFOB filter bandwidth

#define PD 1                    // 1 = enable pseudo
    derivative feedback
#define Gpd 4000                // Filtered derivative
    bandwidth

#define PI 3.14159265358979323846
#define f 1                     // Pulse of sinusoidal
    external force [Hz]

// Environmental properties
#define X_ENV 0.02              // Position of environment
#define Ke 1000                 // Stiffness (Kelvin-Voigt
    model) [N/m]
#define De 100                  // Viscosity (Kelvin-Voigt
    model) [Ns/m]
#define BETA 0.99               // Forgetting factor
```

```
#define K 100000                // Stiffness (Hunt-Crossley
    model)
#define Lambda 10000            // Viscosity (Hunt-Crossley
    model)
#define n 1.2                   // Exponent (Hunt-Crossley
    model)

// Initial Values
#define KINIT 1e3               // Initial value of eK
#define LINIT 1e2               // Initial value of eLambda
#define NINIT 1.5               // Initial value of en
#define GAMMA 100               // Initial value of r11, r22

// Forgetting factor update law coefficients
#define alpha1  0.05
#define alpha2  500
#define alpha3  2

// Switches for estimation mode
#define UPDATE 1                // 0 = no update, 1 = update
#define KL_EST 1                // 0 = basic est, 1 =
    advanced est, 2 = fixed value
#define N_EST 1                 // 0 = basic est, 1 =
    advanced est, 2 = fixed value

// Modifier of sampling frequency
#define COUNT 10                // fc = 1/(T*COUNT)

// Dither frequency
#define fd 150
```

```
double absolute(double value){

    if(value >= 0)
        return value;
    else
        return -value;
}

int main() {

    double time = 0;

    //Common mode and differential mode variables
    double ddx_com_ref = 0, ddx_dif_ref = 0;
    double ddx_com_cmd = 0;
    double ddx_com_res = 0;
    double ddx_dif_res = 0, dx_dif_res = 0, x_dif_res = 0;

    // Force and position control variables
    double Fm_ref = 0, Fs_ref = 0;
    double Fm_res = 0, Fs_res = 0;
    double ddxm_ref = 0, ddxs_ref = 0;
    double ddxm_res = 0, ddxs_res = 0;
    double dxm_res = 0, dxs_res = 0;
    double xm_res = 0, xs_res = 0;
    double ddxm_f = 0, ddxs_f = 0;

    // Force command variables
    double Fm_tot = 0, Fs_tot = 0;
    double Fm = 0;
    double Fs = 0;
```

```
// Variables for update
double ddxm_temp = 0, ddxs_temp = 0;
double dxm_temp = 0, dxs_temp = 0;
double xm_temp = 0, xs_temp = 0;
double dFm_temp = 0, dFs_temp = 0;
double Fm_temp = 0, Fs_temp = 0;

// DOB and RFOB structures variables
double Fm_int = 0, Fs_int = 0;
double eFm_dob = 0, eFs_dob = 0;
double Fm_intr = 0, Fs_intr = 0;
double eFm_ext = 0, eFs_ext = 0;
double eFm_dis = 0, eFs_dis = 0;

// Pseudo-derivative variables
double xm_int = 0, xs_int = 0;
double dxm_int = 0, dxs_int = 0;
double Ddxm_res = 0, Ddxs_res = 0;
double Dddxm_res = 0, Dddxs_res = 0;
double Ddxm_temp = 0, Ddxs_temp = 0;
double Dddxm_temp = 0, Dddxs_temp = 0;

// Environmental response variables
double fKe = 0, fDe = 0;
double deltaX = 0;
double xn = 0;

// Estimation algorithm variables
double eF = 0;
double eF_err = 0;
double en = NINIT;
double beta = BETA;
double mu1 = 1, mu2 = 1;
double q1 = 1, q2 = 1;
double q1t = 0, q2t = 0;
double beta_sum = 0;
double theta1 = 0, theta2 = 0;
double ek = KINIT, elambda = LINIT;
double r11 = GAMMA, r12 = 0, r21 = 0, r22 = GAMMA;
double r11t = 0, r12t = 0, r21t = 0, r22t = 0;
double phi = 0;
double mu = 1;
double q = 1;
double theta = 0;
double r = 0.001;//GAMMA;
double rt = 1;
double err = 0;

// Contact estimation variables
int contact = 0;
int trigger = 0;
double time_c = 0;
double ex_env = 0;
double Fd = 0;

// Sampling counter
int counter = COUNT;

// Output files initialization

char *writenameBF = WRITEFILEBF;
FILE *fwBF;
if((fopen_s(&fwBF, writenameBF, "w")) != 0){
            printf(stderr, "File open error\n");
            exit(1);
    }

fprintf(fwBF, "# 1 time\t2 Fm\t\t3 Fm_res\t4 Fs\t\t5
    Fs_res\t6 F_CMD\t\t7 FMDIS\t\t8 eFm_dob\t9 eFm_dis\
    t10 FSDIS\t11 eFs_dob\t12 eFs_dis\n");

char *writenameBP = WRITEFILEBP;
FILE *fwBP;
if((fopen_s(&fwBP, writenameBP, "w")) != 0){
            printf(stderr, "File open error\n");
            exit(1);
    }

fprintf(fwBP, "# 1 time\t2 xm_res\t3 xs_res\t4 dxm_res\
    t5 dxs_res\t6 Ddxm_res\t7 Ddxs_res\n");

char *writename = WRITEFILE;
FILE *fw;
if((fopen_s(&fw, writename, "w")) != 0){
            printf(stderr, "File open error\n");
            exit(1);
    }

fprintf(fw, "# 1 time\t2 Fm_res\t3 Fs_res\t4 xm_res\t5
    xs_res\t6 Ddxm_res\t7 Ddxs_res\t8 X_ENV\t\t9 deltaX\
    t10 x_env\t11 contact\t12 eFs_ext2\n");

char *writename2 = WRITEFILE2;
FILE *fwe;
if((fopen_s(&fwe, writename2, "w")) != 0){
            printf(stderr, "File open error\n");
            exit(1);
    }

fprintf(fwe, "# 1 time\t 2 eF\t\t 3 Fs_res\t 4 eF_err\t
    5 K\t 6 ek\t 7 Lambda 8 elambda\t 9 n\t\t 10 en\t\t
    11 beta\t\t 12 contact\n");

// Start execution

for(int i = 0; i < STEP; i++){

    time = i*T;

    // CONTROL SYSTEM

    // Reference
    ddx_com_ref = -1*(Kf*ddx_com_res);
    ddx_dif_ref = Ka*ddx_dif_res + Kv*dx_dif_res + Kp*
        x_dif_res;

    ddxm_ref = (ddx_com_ref - ddx_dif_ref)/2;
    ddxs_ref = (ddx_com_ref + ddx_dif_ref)/2;

    Fm_ref = Mn*ddxm_ref + eFm_dob;
    Fs_ref = Mn*ddxs_ref + eFs_dob;

    // Force command
    if(time > 0.25){
        if(trigger){
            Fm = F_CMD*(0.5*cos(2*PI*f*(time - time_c) -
                PI) + 0.6);
        }
    else{
        Fm = 0.001*F_CMD;
        Fd = 0.04*sin(2*PI*fd*time);    // Dither
            force
    }
}
else{
    Fm = 0;
    Fd = 0;
}

// Force reference
Fm_tot = Fm_ref + (Fm + FMDIS) + Fd;
Fs_tot = Fs_ref + (Fs + FSDIS);

// SYSTEM MODEL

ddxm_temp = Fm_tot/Mn;
ddxs_temp = Fs_tot/Mn;

dxm_temp = dxm_res + (ddxm_temp + ddxm_res)*T*0.5;
dxs_temp = dxs_res + (ddxs_temp + ddxs_res)*T*0.5;

xm_temp = xm_res + (dxm_temp + dxm_res)*T*0.5;
xs_temp = xs_res + (dxs_temp + dxs_res)*T*0.5;

// DERIVATIVES

#if(PD)
Ddxm_temp = Gpd*(xm_temp - xm_int);
Ddxs_temp = Gpd*(xs_temp - xs_int);
```

```
xm_int += T*(Ddxm_res + Ddxm_temp)*0.5;
xs_int += T*(Ddxs_res + Ddxs_temp)*0.5;

Dddxm_temp = Gpd*(Ddxm_temp - dxm_int);
Dddxs_temp = Gpd*(Ddxs_temp - dxs_int);

dxm_int += T*(Dddxm_res + Dddxm_temp)*0.5;
dxs_int += T*(Dddxs_res + Dddxs_temp)*0.5;
#endif

// Force from environment

if(xs_temp > X_ENV){
    #if(ENVIRONMENT == 0)       // Kelvin-Voigt
        model
        fKe = Ke;
        fDe = De;

        Fs = -(fKe*(xs_temp - X_ENV) + fDe*(dxs_temp
            ));

    #elif(ENVIRONMENT == 1)     // Hunt-Crossley
        model
        xn = pow(xs_temp - X_ENV, n);

        fKe = K*xn;
        fDe = Lambda*xn;

        Fs = -(fKe + fDe*dxs_temp);

    #endif
}
else{
    Fs = 0;
}

// DISTURBANCE OBSERVER

eFm_dob = Fm_int - Ddxm_temp*Mn*Gobs;
Fm_int  +=      ((Fm_ref + Ddxm_temp*Mn*Gobs) -
    Fm_int)*Gobs*T;

eFs_dob = Fs_int - Ddxs_temp*Mn*Gobs;
Fs_int  +=      ((Fs_ref + Ddxs_temp*Mn*Gobs) -
    Fs_int)*Gobs*T;

// Estimate disturbance when force command is 0
if(time < 0.25){
    eFm_dis = eFm_dob;
    eFs_dis = eFs_dob;
}

// REACTION FORCE OBSERVER

eFm_ext = -(Fm_intr - Ddxm_temp*Mn*Gobs);
Fm_intr +=      ((Fm_ref + Ddxm_temp*Mn*Gobs) -
    Fm_intr - eFm_dis)*Gobs*T;

eFs_ext = Fs_intr - Ddxs_temp*Mn*Gobs;
Fs_intr +=      ((Fs_ref + Ddxs_temp*Mn*Gobs) -
    Fs_intr - eFs_dis)*Gobs*T;

eFs_ext2 = Fs_intr2 - Ddxs_temp*Mn*Greac;
```

```
Fs_intr2        +=      ((Fs_ref + Ddxs_temp*Mn*
    Greac) - Fs_intr2 - eFs_dis)*Greac*T;

// EVALUATE CONTACT

// if(contact > 0 && (eFs_ext + eFm_ext) < 1e-2)
//     contact = 2;

if(time > 0.3 && eFs_ext >= 0.1 && ex_env == 0){
    contact = 1;
    trigger = 1;
    ex_env = xs_temp;
    time_c = time;
}

// if(ex_env != 0)
//     deltaX = xs_temp - ex_env;
if(xs_temp > X_ENV)
    deltaX = xs_temp - X_ENV;

// if(xs_temp > X_ENV && eFs_ext > 0){
if(deltaX > 0 && eFs_ext > 0)
    contact = 1;
else
    contact = 0;

// Estimate environment properties

if(contact){

    // ESTIMATE K, LAMBDA
```

```
mu1 = pow(deltaX, en);
mu2 = mu1*Ddxs_temp;

eF = mu1*ek + mu2*elambda;

eF_err = eFs_ext - eF;

#if(KL_EST==0)
// BASIC ESTIMATION

beta_sum = beta + mu1*(r11*mu1 + r12*mu2) + mu2
    *(r12*mu1 + r22*mu2);

r11 = r11/beta_sum;
r12 = r12/beta_sum;
r22 = r22/beta_sum;

q1 = r11*mu1 + r12*mu2;
q2 = r12*mu1 + r22*mu2;

theta1 = ek + q1*eF_err;
theta2 = elambda + q2*eF_err;

ek = theta1;
elambda = theta2;

#elif(KL_EST==1)
// ADVANCED ESTIMATION

q1t = r11*mu1 + r12*mu2;
q2t = r12*mu1 + r22*mu2;
```

```
beta_sum = beta + mu1*q1t + mu2*q2t;

q1 = q1t/beta_sum;
q2 = q2t/beta_sum;

theta1 = ek + q1*eF_err;
theta2 = elambda + q2*eF_err;

r11t = r11 - r11*q1*mu1 - r12*q1*mu2;
r12t = r12 - r12*q1*mu1 - r22*q1*mu2;
r22t = r22 - r12*q2*mu1 - r22*q2*mu2;

r11 = r11t/beta;
r12 = r12t/beta;
r22 = r22t/beta;

ek = theta1;
elambda = theta2;
#endif

// ESTIMATE N

#if(N_EST==0)
// BASIC ESTIMATION

phi = log(eFs_ext/(ek + elambda*Ddxs_temp));
mu = log(deltaX);

r = r/(beta + mu*r*mu);

q = r*mu;
```

```
    theta = en + q*(phi - mu*en);
    if(theta < 1)
        theta = 1;

    en = theta;
    #elif(N_EST==1)
    // ADVANCED ESTIMATION

    phi = log(eFs_ext/(ek + elambda*Ddxs_temp));
    mu = log(deltaX);

    q = r*mu/(beta + mu*r*mu);

    err = phi - mu*en;

    theta = en + q*err;
    if(theta < 1)
        theta = 1;

    r = (r - r*q*mu)/beta;

    en = theta;
    #endif

    #if UPDATE
    beta = 1 - alpha1*(0.5 + (atan(alpha2*(absolute(
        eF_err) - alpha3)))/PI);
    #endif
}

// Update variables
ddxm_res = ddxm_temp;
```

```
ddxs_res = ddxs_temp;
dxm_res = dxm_temp;
dxs_res = dxs_temp;
xm_res = xm_temp;
xs_res = xs_temp;
Fm_res = eFm_ext;
Fs_res = eFs_ext;

#if(PD)
Dddxm_res = Dddxm_temp;
Dddxs_res = Dddxs_temp;
Ddxm_res = Ddxm_temp;
Ddxs_res = Ddxs_temp;
#endif

// CALCULATE ACCELERATION

// Force loop

ddxm_f = -Fm_res/Mn;
ddxs_f = Fs_res/Mn;

ddx_com_res = ddxm_f + ddxs_f;

// Position loop

#if(PD)
ddx_dif_res = Dddxm_res - Dddxs_res;
dx_dif_res = Ddxm_res - Ddxs_res;
#elif(!PD)
ddx_dif_res = ddxm_res - ddxs_res;
dx_dif_res = dxm_res - dxs_res;
```

```
#endif
x_dif_res = xm_res - xs_res;



if(counter < COUNT){
    counter++;
}
else{
    // Write in output file

    fprintf(fwBF, "%f \t%f \t%f \t%f \t%f \t%f \t%f
        \t%f \t%f \t%f \t%f \t%f\n",
    time, Fm, Fm_res, Fs, Fs_res, F_CMD, FMDIS,
        eFm_dob, eFm_dis, FSDIS, eFs_dob, eFs_dis);

    fprintf(fwBP, "%f \t%.10f \t%.10f \t%.10f \t%.10
        f \t%.10f \t%.10f\n",
    time, xm_res, xs_res, dxm_res, dxs_res, Ddxm_res
        , Ddxs_res);

    fprintf(fw, "%f\t %f\t %f\t %.10f\t %.10f\t %.10
        f\t %.10f\t %f\t %f\t %f\t %d\t %f\n",
    time, Fm_res, Fs_res, xm_res, xs_res, Ddxm_res,
        Ddxs_res, X_ENV, deltaX, ex_env, contact,
        eFs_ext2);

    fprintf(fwe, "%f\t %f\t %f\t %f\t %d\t %f\t %d\t
         %f\t %f\t %f\t %f\t %d\n",
    time, eF, Fs_res, eF_err, K, ek, Lambda, elambda
        , n, en, beta, contact);

    counter = 1;
```

```
        }

        }

    fclose(fw);
    fclose(fwe);
    fclose(fwBF);
    fclose(fwBP);

    return 0;

}
```

BIBLIOGRAPHY

[1] M. Li, H. Yin, K. Tahara, and A. Billard. Learning object-level impedance control for robust grasping and dexterous manipulation. In *2014 IEEE International Conference on Robotics and Automation ICRA)*, pages 6784–6791, 2014.

[2] N. Hogan. Stable execution of contact tasks using impedance control. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 1047–1054, 1987.

[3] P. Paoletti, G. Jones, and L. Mahadevan. Grasping with a soft glove: Intrinsic impedance control in pneumatic actuators. *Journal of The Royal Society Interface*, 14, 03 2017.

[4] R. Ma, A. Spiers, and A. Dollar. M2 gripper: Extending the dexterity of a simple, underactuated gripper. volume 36, 07 2015.

[5] T. Shimono, S. Katsura, and K. Ohnishi. Abstraction and Reproduction of Force Sensation From Real Environment by Bilateral Control. *IEEE Trans. on Industrial Electron.*, 54 (2), April 2007.

[6] Y. Asai, Y. Yokokura, and K. Ohishi. Fine Force Reproduction of Environmental Haptic Sensations Based on Momentum Control. *IEEE Transactions on Industrial Electronics*, 63(7):4304–4313, 2016.

[7] T. Shimono, S. Katsura, R. Kubo, and K. Ohnishi. Multilateral Control for Skill Education based on Haptic Data Storage. In *2006 IEEE International Conference on Industrial Technology*, pages 334–339, 2006.

[8] H. Muramatsu and S. Katsura. Repetitive Motion-Rreproduction Based on a Motion-Copying System for Automation of Human Motions. In *SICE Annual Conference 2017*, Kanazawa, Japan, 2017.

[9] A. Pappalardo, A. Albakri, C. Liu, L. Bascetta, E. De Momi, and P. Poignet. Hunt-Crossley Model Based Force Control For Minimally Invasive Robotic Surgery. *Biomedical Signal Processing and Control*, 29, 05 2016. doi: 10.1016/j.bspc.2016.05.003.

[10] L. J. Love and W. J. Book. Environment estimation for enhanced impedance control. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 2, pages 1854–1859 vol.2, 1995.

[11] Y. Asai, Y. Yokokura, and K. Ohishi. Mechanical Admittance Realization Control Based on Feedforward Compensation for Fine Realization of Impedance. *IEEJ Journal of Industry Applications*, 6(2):83–90, 2017.

[12] Lewis FL, Dawson DM, and C Abdallah. *Robot Manipulator Control: Theory and Practice*. 01 2004.

[13] N. Diolaiti, C. Melchiorri, and S. Stramigioli. Contact Impedance Estimation for Robotic Systems. *IEEE Trans. on Robotics*, 21(5), October 2005.

[14] A. Haddadi and K. Hashtrudi-Zaad. Real-Time Identification of Hunt-Crossley Dynamic Models of Contact Environments. *IEEE Trans. on Robotics*, 28(3), June 2012.

[15] R. Shindeler and K. Hashtrudi-Zaad. Online Identification of Environmental Hunt-Crossley Models Using Polynomial Linearization. *IEEE Trans. on Robotics*, 34(2), April 2018.

[16] Z. Wang, A. Peer, and M. Buss. Fast online Impedance Estimation for Robot Control. In *2009 International Conference on Mechatronics*, Malaga, Spain, April 2009.

[17] K. Tanida, K. Ohnishi, T. Mizoguchi, D. Yashiro, Y. Morikawa, and N. Shimojima. Measuring Stiffness of Organs using 2-DOF Haptic Surgical Forceps Robots. In *2011 1st International Symposium on Access Spaces (ISAS)*, pages 242–247, 2011.

[18] M. Mizuochi and K. Ohnishi. Contact Detection using Dither in Force Sensorless Motion Control. In *2013 IEEE International Conference on Mechatronics (ICM)*, pages 254–259, Vicenza, Italy, 2013.

[19] A. Matsui, Y. Yokokura, and S. Katsura. Dynamic Data Reproduction of Contact Environment with State Transition. *IEEJ Journal of Industry Appl.*, 2(1):22–29, 2013.

[20] T. Nozaki, T. Mizoguchi, and K. Ohnishi. Real-World Haptics for Motion Realization. *IEEJ Trans. on Industrial Electron.*, 2(1):1–6, 2013.

[21] H. Kazerooni, T. I. Tsay, and C. L. Moore. Telefunctioning: An approach to telerobotic manipulations. In *1990 Amer-*

ican Control Conference, pages 2778–2783, May 1990. doi: 10.23919/ACC.1990.4791228.

[22] Mehmet Dede and Sabri Tosunoglu. Fault-tolerant teleoperation systems design. *Industrial Robot: An International Journal*, 33:365–372, 09 2006.

[23] Weston Blaine Griffin. *Shared Control For Dexterous Telemanipulation With Haptic Feedback*. 2003.

[24] Dongjun Lee and M. W. Spong. Bilateral teleoperation of multiple cooperative robots over delayed communication networks: Theory. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 360–365, 2005.

[25] Y. Yokokohji and T. Yoshikawa. Bilateral control of master-slave manipulators for ideal kinesthetic coupling. In *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, pages 355–362 vol.1, 1990.

[26] B. Hannaford. A design framework for teleoperators with kinesthetic feedback. *IEEE Transactions on Robotics and Automation*, 5(4):426–434, 1989.

[27] K. Jayarajan and S. Manjit. Master-slave manipulators: Technology and recent developments. *BARC News Letter*, 269:2–12, 06 2006.

[28] Thomas Sheridan and William L. Verplank. Human and computer control of undersea teleoperators. 07 1978.

[29] Surveyor 3. https://en.wikipedia.org/wiki/Surveyor_3.

[30] Office of Scientific NASA and Technical Information. *Scientific and Technical Aerospace Reports*, volume 25. 1987.

[31] Viking program. https://en.wikipedia.org/wiki/Viking_program.

[32] Spar Aerospace Ltd Lydia Dotto, David A. B. Steel. *A Heritage of Excellence: 25 Years at Spar Aerospace Limited*. 1992.

[33] Canadarm. https://en.wikipedia.org/wiki/Canadarm.

[34] Flint Wild. What is robonaut? https://www.nasa.gov/audience/forstudents/5-8/features/nasa-knows/what-is-robonaut-58.html.

[35] Woo-Keun Yoon, Toshihiko Goshozono, Hiroshi Kawabe, Masahiro Kinami, Yuichi Tsumaki, Masaru Uchiyama, Mitsushige Oda, and Toshitsugu Doi. Model-based space robot teleoperation of ets-vii manipulator. *Robotics and Automation, IEEE Transactions on*, 20:602 – 612, 07 2004. doi: 10.1109/TRA.2004.824700.

[36] T. Imaida, Y. Yokokohji, T. Doi, M. Oda, and T. Yoshikawa. Ground-space bilateral teleoperation of ets-vii robot arm by direct bilateral coupling under 7-s time delay condition. *IEEE Transactions on Robotics and Automation*, 20(3):499–511, 2004.

[37] M. Cenk Cavusoglu. Telesurgery and surgical simulation: Design, modeling, and evaluation of haptic interfaces to real and virtual surgical environments. Aug 2000. URL http://www2.eecs.berkeley.edu/Pubs/TechRpts/2000/3874.html.

[38] S.E. Butner and Moji Ghodoussi. Transforming a surgical robot for human telesurgery. *Robotics and Automation, IEEE Transactions on*, 19:818 – 824, 11 2003.

[39] Metin Sitti and Hideki Hashimoto. Teleoperated touch feedback from the surfaces at the nanoscale: Modeling and experiments. *Mechatronics, IEEE/ASME Transactions on*, 8: 287 – 298, 07 2003.

[40] S. Khan, A. Sabanovic, and A. O. Nergiz. Scaled bilateral teleoperation using discrete-time sliding-mode controller. *IEEE Transactions on Industrial Electronics*, 56(9):3609–3618, 2009.

[41] K. Miura, A. Matsui, and S. Katsura. High-Stiff Motion Reproduction Using Position-Based Motion-Copying System With Acceleration-Based Bilateral Control. *IEEJ Trans. on Industrial Electron.*, 62(12), December 2015.

[42] N. tsunashima and S. Katsura. Spatiotemporal Coupler: Storage and Reproduction of Human Finger Motion. *IEEE Trans. on Industrial Electron.*, 59(2), February 2012.

[43] G. Gilardi and I. Sharf. Literature survey of contact dynamics modeling. *Mechanism and Machine Theory*, 37:1213–1239, 10 2002.

[44] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. 01 2011.

[45] H. Hertz, D. E. Jones, and G. A. Schott. *Miscellaneous papers*. London, 1896.

[46] W. Goldsmith. *Impact: The Theory and Physical Behavior of Colliding Solids*. Edward Arnold Publishers Ltd, London, 1960.

[47] Luka Skrinjar, Janko Slavic, and Miha Boltezar. A review of continuous contact-force models in multibody dynamics. *International Journal of Mechanical Sciences*, 145:171–187, 09 2018.

[48] S. Dubowsky and F. Freudenstein. Dynamic analysis of mechanical systems with clearances, part 1: Formation of dynamical model. *Journal of Engineering for Industry*, 93: 305–309, 1971.

[49] F.R.E. Crossley K.H. Hunt. Coefficient of restitution interpreted as damping in vibroimpact. *Journal of Applied Mechanics*, 42, Series E:440–445, 1975.

[50] R. G. Herbert and D. C. McWhannell. Shape and frequency composition of pulses from an impact pair. *Journal of Engineering for Industry*, 99(3):513–518, 1977.

[51] A.C. Wang T.W. Lee. On the dynamics of intermittent-motion mechanisms, part 1: Dynamic model and response. *Journal of Mechanisms, Transmissions, and Automation in Design*, 105:534–540, 1983.

[52] H. M. Lankarani and P. E. Nikravesh. A contact force model with hysteresis damping for impact analysis of multibody systems. *Journal of Mechanical Design*, 112(3): 369–376, 1990.

[53] D. W. Marhefka and D. E. Orin. A compliant contact model with nonlinear damping for simulation of robotic systems.

IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 29(6):566–572, 1999.

[54] Lennart Ljung and Torsten SÃ¶derstrÃ¶m. *Theory and Practice of Recursive Identification*. 1983.

[55] S. Bittanti, Marco Campi, and Lei Guo. Persistence of excitation properties for the identification of time-varying systems. volume 1, pages 680–684, 02 2000.