Università degli Studi di Padova

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI

Corso di Laurea Magistrale in Ingegneria Meccatronica

Tesi di Laurea Magistrale

Bilateral Control with Task Learning and Adaptation to Environment

Candidato:
Marco Cavazza

Relatore:
Prof. Roberto Oboe

Anno Accademico 2018–2019

ABSTRACT

The topic of this thesis is the study of techniques for the recording of user motions as accurately as possible, with the intent of saving a digital copy of this knowledge. While other forms of expertise or creativity can be registered with photos or audio/video recordings, physical actions cannot: assuming that the user uses a tool, both the position and the applied force are needed to effectively record the action. The user motion recording could be useful in Countries with a big population gap between two generations, that will lead in the coming decade to a shortage of workers: the saved copy can then be used to train new operators or artificially reproduce the saved task, without needed the presence of a person.

A technology being studied to save an operator task is the Motion Copying System, that is based on the Bilateral Control. Because of the need to save both positions and forces, the user doesn't directly interact with the environment, instead, two actuators are used to mirror the user actions. A first actuator is used to operate on the environment, while the user interacts with the second one: position and forces are mirrored between the two, so that the user feels haptic feedback from the environment, improving the tactile sensitivity. The main drawback of this solution is that the saved action is rigid with respect to environmental changes. The saved position and force refers to a specific environmental configuration: if the target position or mechanical properties vary, the saved motion isn't compatible anymore and the reproducing the task will yield position and force errors.

This work proposes to model each operator action as a motion primitive that can be easily adapted to the environment changes, with the future objective of saving a generalized version of each action to a database, from which a task can be algorithmically generated without the need of user intervention. The experimental setup used in this thesis uses a basic 1-DOF mechanical system, that although verifies the proposed solution, constitutes the main limitation of this work.

ACKNOWLEDGEMENT

In the first place I would like to express my sincere gratitude to the two Professors that helped me realize this thesis: my supervisor, Prof. Oboe of the University of Padova, for the help he offered, the numerous inputs for the improvement of my work, the patience and the always welcomed friendly small talk, and Prof. Shimono of the Yokohama National University, for the life-changing opportunity he offered me, to stay 6 months in Japan, welcoming me to his laboratory and being always helpful, friendly and supportive.

I would like to thank all the students at Shimono Laboratory at YNU, for the wholesome experience, the welcoming environment and the opportunity to live in a different country from what I was used to.

ありがとうございます – Sincerely, Thank You.

I thank my family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Finally I would like to thank my friends, both from Italy and from Japan. In particular thank to Edoardo, Andrea V. F. Z., Giuseppe, Fabio, Francesco, Stella, Cle, Demo, Mattia and Marco for their support.

Grazie.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 GLOBAL POPULATION AGEING AND FUTURE WORKERS SHORTAGE

The world-wide improvement of the life expectancy during the last one and a half century and the declining fertility rates are well-established phenomena, that result in a combined effect of progressive population aging. The life expectancy in Asia, for example, doubled in less than 100 years (Figure 1), while the world-average fertility rate (number of children per woman) halved during the last 65 years (Figure 2). This process is going to be even more fast-paced for currently developing countries.

Ignoring the deep sociological effects caused by this phenomenon, many questions arise about the ability of the industrial manufacturing of being able to sustain the current level of production, when the present workforce will retire and fewer and fewer workers will be available to fill the empty positions. The industrial processes automation represents a solution to the economy of scale production, where low-skilled workers (that are going to be increasingly expensive) can be substituted by ad-hoc automatic machines or production lines, with the benefits of relieving those workers from heavy, boring or dangerous tasks. However, this solution overlooks the highly-skilled low-volume activities typical, for example, of the artisan production. A portion of these specific activities can be automated through the use of robots that are more flexible than production lines with respect to the task they are assigned to. Anthropomorphic 7 degrees-of-freedom robots can mimic the actions accomplished by a human arm, like welding along a complicated path. In the last several years, the introduction of collaborative robots permits to safely mix operators and machines in the same workspace.

Even with the use of robots, some tasks are difficult to automate because some degree of craftsmanship is required in the production process. This usually involves some manual expertise that is accumulated during a long time of practice and is not easily scalable in the production because new employees need training, which takes time. With the aforementioned future shortage of workers, the problem of keeping up with the production and safeguarding the current employees' knowledge is emerging. The video recording of the task execution is usually not sufficient to reconstruct the full action, because it lacks the force information exerted by the operator: the system we are looking for should ideally be *interposed between the operator and the environment*, while being *transparent to the user*, recording the forces and movements associated with the task and then being able to re-enact the recorded motion with the increased sensibility.

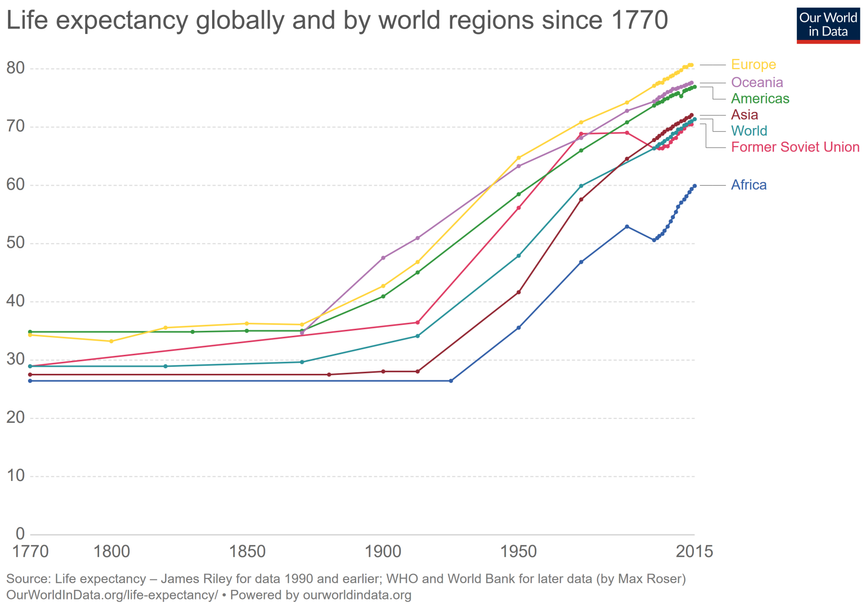


Figure 1: Global life expectancy since 1770
ourworldindata.org/life-expectancy

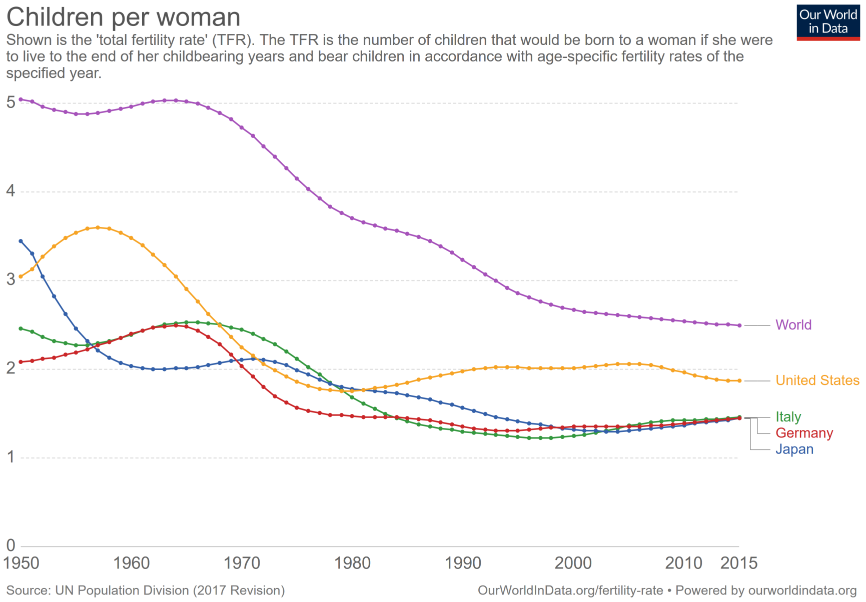


Figure 2: Fertility rate since 1950
ourworldindata.org/fertility-rate

1.2 MOTION COPYING SYSTEM

A **Motion Copying System** (MCS) is based on a bilateral control system, that is composed by a master and a slave actuators: the user interacts with the first, while the second is located at the worksite. As the operator acts on the master, the movement is replicated ad the slave side. To increase the user sensibility, the bilateral control provides haptic feedback by mirroring the applied forces: the force exerted by the user is applied to the environment and the reaction force is mirrored back to the operator.

The Motion Copying System works by:

- during the operator action the system perform as a bilateral system, but the master position, speed and force signals are recorded to a *motion database*;
- using the saved references as a *virtual master* the action can be reproduced whenever needed.

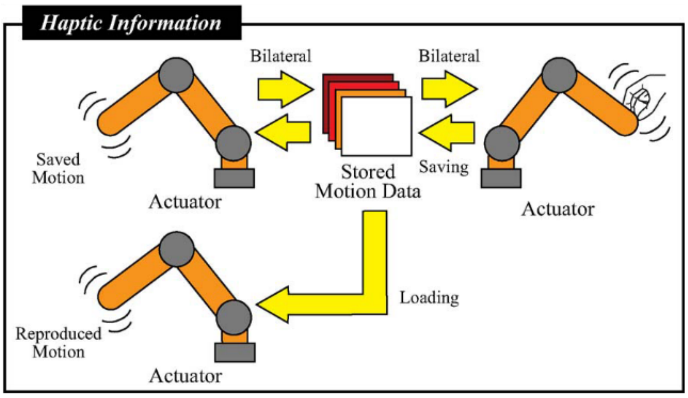


Figure 3: Motion Copying System principle operation

If the action is replayed when the environment conditions didn't change from the recording session the applied position and force match the operator's, but if the conditions change, because the virtual master is "read-only", the action will present position and force errors. Recording the position and force signals is equivalent to memorizing the environment impedance (stiffness and damping) so, if the actual impedance changes from the recording time, there will be a mismatch between the recorded and the actual environment conditions. As will be presented, the bilateral system feed by the recorded coordinates is still stable, but this mismatch will cause a position and force errors that compromises the precise task reproduction.

The simplest and most effective solution is to record a new version of the task; this isn't a long-term viable solution, because still needs the operator intervention, even if not as frequently as before. The ideal goal is to "learn" a generalized version of the task, that can be easily adapted to new operating conditions.

The MCS is derived from a **bilateral control system**, that was originally developed to realize a teleoperation devices, enabling the operator to interact with the environment remotely. An important aspect of teleoperation systems is the degradation of the system stability as the communication delay (*lag*) increases. The control theory behind such systems is extensively studied and working apparatuses have been successfully realized and deployed. These systems, originally developed for teleoperation, can be used on a more local setting by scaling the operator movement and force amplitude to lighten the workload, but still requires the presence of the user.

1.3 TASK MODELLING

In the literature several options are available to model a MCS task.

1.3.1 Neural Networks

Artificial Neural Networks (ANNs) are inspired by biological neural networks, composed by a collection of connected units (*neurons*) that reassemble biological neurons: each connection transmit a signal from one unit to the next ones. In general, each neuron output applies a non-linear function to a weighted combination of other neurons outputs.

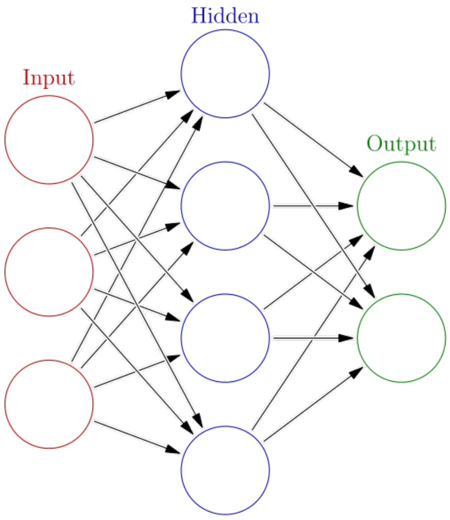


Figure 4: Basic ANN diagram

The training of the network is supervised: the connection weights (arrows in Figure 4) are adjusted by back-propagating the errors of a training set. This framework is very flexible and, by increasing the number of neurons (up to hundreds of thousands) and using different network structures, it can be adapted to many different tasks, from image recognition to natural language processing. In the scientific literature several researches [1, 2] presented different methods to apply ANNs to robot control. The main drawback of this framework is the fundamental role of the resource intensive training phase and the black-box nature of its functioning.

The use of neural networks to represent a motion copying system task was initially disregarded, as the framework complexity and unknowable inner working are not appropriate to model a motion copying system task, for example to simply asses the overall system stability.

1.3.2 Reinforcement Learning

The Reinforcement Learning (RL) is an area of machine learning that design systems that autonomously interact with the environment to maximize a cumulative reward. It differs from the supervised algorithms (e.g. ANNs) because it doesn't need a labeled dataset: instead, given a set of conditions and a set of actions it can execute, it aims to find the sequence of actions that yield the overall best solution, the one that maximizes the cumulative reward. An example reinforcement learning robot control application is presented in [3]. This framework is then not used to model an operator action, but to generate new tasks from specific requirements: this could be useful to adapt a saved action to a different setting.

1.3.3 Dynamic Movement Primitives

The Dynamic Movement Primitives were introduced as a framework for robot control by [4]. The trajectory information for each robot DOF is stored as a set of weights of N radial basis functions (RBFs) equally spaced along the task duration. By increasing the number of RBF, N, a higher density of RBF can model the task with more precision. The function that approximates the original trajectory by combining the RBF with the relative weights is defined *forcing term*.

During the task execution the trajectory information can be scaled both in the amplitudes and in the time dimension (to slow down or speed up the task execution). The *goal* is defined as a scalar value that represent the final state of a DOF at the task completion. The amplitude scaling is obtained by multiplying the forcing term by the goal.

A transformation dynamic system translates the trajectory information to the motor controller references. To do so a second-order dynamic system with point attractor behavior, towards the task goal, is fed with the goal-scaled forcing term.

In the thesis, the DMP framework is adapted to model a motion copying system task by including a second set of weights for each action DOF referred to the force information.

A saved task can be adapted to new environmental conditions by adjusting the position and force goals. This is done following an exploration run to determine how the conditions changed from the original ones.

This DMP framework solution is chosen because

- require only one operator task execution to capture the necessary information

- has a simple structure, easily interpretable (contrary to neural networks)

- the simple structure also influences the amount of computational power needed to implement this solution;
- the task adaptation mechanism is rather simple, two "knobs" to control each motion copying system action position and forces and a time parameter so control the execution speed, that means:
  - the structure is rigid compared to other solutions
  - very straightforward to design and implement new adaptation strategies

The thesis is organized as follows:

- in the second chapter the Motion Copying System is firstly presented, then the bilateral control theory is explained and applied to the mechanical system;
- in the third chapter a Kalman Filter is designed and tuned to estimate the actuators speed and external forces, a extended friction model is presented to avoid the discontinuity of the simple static and viscous friction model
- in the forth chapter the DMP framework is explained, extended to the bilateral control, several adaptation policies presented and tested; an adaptation algorithm based on the Iterative Learning Control is discussed and tested.
- after the thesis conclusions, in the appendix A, the bilateral control reference scaling properties are briefly presented and tested.

2 MOTION COPYING SYSTEM

2.1 INTRODUCTION

In this chapter a Motion Copying System (MCS) will be presented, designed and realized, both in simulation and on the experimental bench. The goal of a MCS is to save an operator task by recording the position and applied force of an user-controlled end-effector. To accomplish this, the MCS uses a Bilateral Control system, that permits to an operator to indirectly perform a task on the environment by interacting with a master robot, while another robot performs the actual action, receiving force feedback from the environment without the constraint for the operator of being in the same location.

This system allows the operator to work from a more comfortable or safer location, decoupling the operator position from the location of the task action.

As briefly introduced above, a further step is achieved by decoupling the action recording *time* from the task reproduction time thanks to the Motion Copying System, this way a task can be recorded and used when the operator is not available or to train other people.

A Motion Copying System (MCS, [5]) uses an existing bilateral control system to save the operator task in the form of the master position and applied force references, that can be used to "replay" the operator task at a different time.

The ability to digitally save a human action might be valuable in different situations. In the first place, it can be useful to preserve in a digital form a specific motion that otherwise will be lost if the person who performs it isn't able to execute it anymore. Differently from traditional robot control, the position reference alone is not enough to re-enact the original motion executed by the operator.

If the action is repetitive or it needs to be performed at a particular time, a possible solution to avoid the constant presence of an operator is to record the action (or several) and "replay" it when necessary. Because this system can copy and reproduce the operator actions it was introduced under the *Motion Copying System* name [5].

2.2 BILATERAL CONTROL BACKGROUND

In the industrial environment, a common operator task is to perform a certain activity involving the interaction with the surroundings – using usually a dedicated tool – that requires some degree of precision, for example handling a fragile object, cutting or welding. In some cases these operations need to be performed under challenging environmental conditions, such as high temperatures near a furnace or in the proximity of hazardous materials. While many repetitive or dangerous tasks can be automated, there is a subgroup where traditional robots cannot be employed because the applied fine force sensibility of the operator is required.

The Bilateral Control was introduced from the necessity to spatially decouple the working apparatus from the operator: this objective is achieved by letting the user control a tool replacement/surrogate (*master* side) that mirrors the user input to the actual end-point (*slave* side). The innovative feature of the bilateral control is the ability to provide environmental force feedback (haptic feedback) between the master and slave system, with great benefit from user operation sensibility, who can remotely feel the environmental conditions and act accordingly, as if the user was operating in first person. This feature requires:

- an *active* master side, capable of generating the environment reaction force to the user;
- the ability to measure the applied forces to the master side by the user and to slave by the environment reaction force.

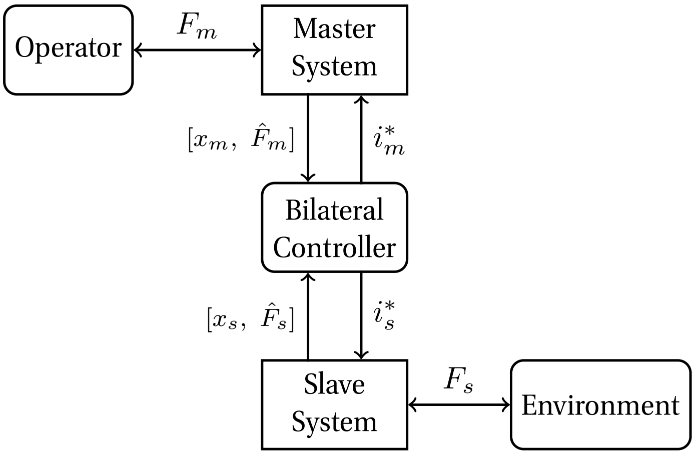


Figure 5: Bilateral Control structure

To accomplish the second requirement, the most simple and immediate solution is to use force sensors mounted on both master and slave systems. Although possible, this would result in an added cost and, depending on the sensor construction technology, have several limitations, mainly the available sensor bandwidth and the presence of measurement bias. Instead, the total external forces acting on each motor are estimated by a disturbance observer (DOB); a DOB uses the motor model and current reference to estimate the total disturbance acting on it. During the bilateral operation the user handles the master robot while the controller mirrors positions and forces between the two systems, digitally re-enacting the law of action-and-reaction as if the two were mechanically linked.

As the slave robot interacts with the environment, the external forces are estimated and mirrored to the operator by driving the master robot accordingly. The user can now act appropriately, based on the field experience he already has; coupled with an video stream, this setup permits to the operator to work as if he was operating on the slave environment himself.

Moreover, because the position and force information is now handled by a digital controller, it is possible to scale the position and forces between master and slave, e.g. amplify the slave actuator force to lighten the forces the operator need to exert or amplify the master position and speed to increase the operator precision and sensibility. Applications of the bilateral control can be multiple:

- for safety reasons and the secure handling of hazardous materials, for example in chemical plants;
- so the operator can work from a more comfortable / ergonomic position;
- because of the distance between the expert and the working place (i.e. teleoperation),
- because the position and force signals can be scaled by the bilateral controller it is possible to amplify the operator force or to improve the operator precision

Until now the communication time delay between the two robots and the bilateral controller was neglected: if the bilateral controller is directly connected to both robots, generally the time delay can be ignored, because usually much smaller than the sampling period. However, if the communication between the two system happens over some network protocol (e.g. the TCP/IP stack), the performance of the bilateral controller system degrades as the time delay increases: the TCP/IP protocol packet delivery timing is not deterministic, therefore the communication delay is not constant but varies with the network congestion. The effects of the communication delay can range from an inaccurate feedback signal up to the destabilization of the bilateral system when the lag between master and slave interferes with the control stability. In those cases the control robustness can be improved with a time delay compensation: a possible solution is to implement a Communication Disturbance Observer (CDOB [6]) to treat the time delay as an external disturbance that will be compensated at the system input.

2.3 BILATERAL CONTROL THEORY

The design of a bilateral system control structure, that have to achieve both the position tracking and the force mirroring between the master and the slave robot, is not a trivial task: the two control objectives have antagonistic requirements. In fact the position tracking drives the two robots towards the same direction to reduce the position error, while the force mirroring in the opposite one to achieve the null resulting force. The first control objective, the position and speed tracking, can be formulated as:

$$\begin{aligned} e_p &= x_m - x_s \rightarrow 0 \\ e_v &= \dot{x}_m - \dot{x}_s \rightarrow 0 \end{aligned}$$

The second control objective is the force mirroring: the controller drives the motors to obtain the external force balance between the master and slave robots:

$$e_f = F_m + F_s \rightarrow 0$$

If the control objectives are achieved, the whole bilateral system acts like a solid body of infinite stiffness: from the master side it is possible to have perfect tactile feedback from the environment at the slave side as if the two were mechanically linked.

The design of the bilateral control system is done in the *modal space*: given the two bilateral system states (*joint space*) they can be decomposed into the *common* and *differential* states of the modal space. The conversion from the actuator joint space into the modal space is done using the *Quarry* matrix $Q_n \in \mathbb{R}^{n \times n}$ [5], where the first row represents the common mode and the other rows highlights each of the $n-1$ differential modes:

$$Q_n = \frac{1}{n} \begin{bmatrix} 1 & \dots & 1 & 1 & 1 & 1 & 1 \\ 0 & \dots & 0 & 0 & 0 & 1 & -1 \\ 0 & \dots & 0 & 0 & 2 & -1 & -1 \\ 0 & \dots & 0 & 3 & -1 & -1 & -1 \\ 0 & \dots & 4 & -1 & -1 & -1 & -1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ n-1 & \dots & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

The second-order quarry matrix is defined as

$$Q_2 = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad Q_2^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{1}$$

and it is used for the modal decomposition in the bilateral controller inputs, to transform the joint space coordinates (that is each DOF actuators positions, velocities, forces) into the modal space used for the control of the bilateral system

$$\begin{bmatrix} x_c \\ x_d \end{bmatrix} = Q_2 \begin{bmatrix} x_m \\ x_s \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} F_c \\ F_d \end{bmatrix} = Q_2 \begin{bmatrix} F_m \\ F_s \end{bmatrix} \tag{3}$$

where the subscript $\bigcirc_c$ denotes the common mode of the bilateral system and $\bigcirc_d$ the differential mode and, being Q_2 non-singular, it is possible to reconstruct the joint space coordinates with the inverse matrix Q_2^{-1}. The control objectives in the new coordinates are:

$$\begin{aligned} x_d^{ref} &= 0 \\ \dot{x}_d^{ref} &= 0 \\ F_c^{ref} &= 0 \end{aligned}$$

The first two equations explicit the position and speed tracking between the master and slave whereas the last equation imposes that the sum of the master and slave-side forces is null to realize the law of action-reaction between the two environments.

2.3.1 Acceleration-based Bilateral Controller

In the literature [8, 7] the control of the bilateral system is performed in the acceleration dimension (ABC, Acceleration-based Bilateral Control) by the combined action of two controllers: a proportional/derivative position controller and a proportional force controller:

$$\ddot{x}_d^{ref} = -C_p(s)\, x_d \tag{4}$$

$$\ddot{x}_c^{ref} = -C_f(s)\, F_c \tag{5}$$

with

$$C_p(s) = K_p + K_v s \tag{6}$$

$$C_f(s) = K_f \tag{7}$$

where C_p is the position controller, C_f the force controller and s the Laplace operator. The outputs of the two controllers are acceleration references:

- the position controller drives the two systems towards the same direction (differential mode) to reduce the position error between master and slave;
- the force controller drives the two systems in the same direction (common mode) to exert the same force on the environment and on the operator.

The acceleration references in the modal space can be then transformed with the inverse quarry matrix Q_2^{-1} back into the joint coordinate space:

$$\begin{bmatrix} \ddot{x}_m^{ref} \\ \ddot{x}_s^{ref} \end{bmatrix} = Q_2^{-1} \begin{bmatrix} \ddot{x}_c^{ref} \\ \ddot{x}_d^{ref} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \ddot{x}_c^{ref} \\ \ddot{x}_d^{ref} \end{bmatrix} \tag{8}$$

The specific bilateral control mechanical system used in this work is driven by linear motors; the current references i_m^{ref}, i_s^{ref} can be calculated from the acceleration references using the mass matrix and the nominal linear motor force/current constant K_t,

$$\begin{bmatrix} i_m^{ref} \\ i_s^{ref} \end{bmatrix} = \begin{bmatrix} M_m/K_{t,m} & 0 \\ 0 & M_s/K_{t,s} \end{bmatrix} \begin{bmatrix} \ddot{x}_m^{ref} \\ \ddot{x}_s^{ref} \end{bmatrix} \tag{9}$$

where M_m is the equivalent mass of the master robot and M_s of the slave robot; $K_{t,m}$, $K_{t,s}$ are the force/current constants of the linear motors. The controller outputs are acceleration references that are converted to the current references using the motor nominal parameters (mass M_n and the force constant $K_{t,n}$): this further step is useful to maintain the controller design symmetrical between master and slave when the master and slave actuators differ, for example when the moving mass or K_t are different.

This control structure enables the position/speed or force scaling between master and slave: this option is studied in Appendix A.

The experimental test bench in use consists of two master/slave mechanical systems that have matching characteristics and therefore have the same nominal parameters. If this condition is verified then $M_m = M_s = M_n$ is the nominal motor moving mass and $K_{t,M} = K_{t,S} = K_{t,n}$ the nominal force/current constant:

$$\begin{bmatrix} i_m^{ref} \\ i_s^{ref} \end{bmatrix} = \frac{M_n}{K_{t,n}} \begin{bmatrix} \ddot{x}_m^{ref} \\ \ddot{x}_s^{ref} \end{bmatrix} \tag{10}$$

While a properly designed position controller yields a stable system, the force controller alone is not able to guarantee the system stability: a mismatch in the environmental conditions between master and slave could generate an unstable behavior. Using a controller structure that includes the mixed action of both position and force controls permits (after the proper tuning) to benefit from the stable position tracking of the first and the force mirroring of the second; the overall controller won't have the same performance in the single objective as each controller taken individually, but it is able to achieve the bilateral operation with a high degree of functionality. The prominence of the two controllers is chosen by the relative value of the position/velocity controller and the force controller gains. Given that the position controller gain values K_p, K_v are chosen:

- with $K_f = 0$ the system tracks the master and slave position, while the force mirroring is not active;
- for "smaller" K_f values the force mirroring objective is secondary to the position tracking;
- for "bigger" K_f values the force reproduction performance is improved and the position tracking objective is subordinated to the force mirroring.

2.3.2 Environment Impedance Reproduction

The relationship between master and slave positions and forces can described by the **hybrid parameters** of the H matrix [9, 11, 10]. Accounting for the movement directions:

$$\begin{bmatrix} F_m \\ x_m \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} x_s \\ -F_s \end{bmatrix} \tag{11}$$

where x is the position, F the external force, the subscript indicates $\bigcirc_m$ the master side, $\bigcirc_s$ the slave side. From the master-slave position and force relations, the meaning of the H_{ij} terms can be inferred:

- H_{11} represents the impedance between the slave position and the master force
- H_{12} the *force scaling* between master and slave systems
- H_{21} the *position scaling* between master/slave
- H_{22} the impedance between the master position and the slave force

Having defined the bilateral controller structure it is possible to study how the environment impedance is reproduced at the master side. The environment (slave) mechanical impedance Z_s is modeled as:

$$Z_s = \frac{F_s}{x_s} = K_s + D_s\, s \tag{12}$$

where K_s is the stiffness and D_s the damping of the object in contact with the slave robot end-effector. Combining (12) and (11) it is possible to derive the expression for the reconstructed slave impedance at the master side:

$$\begin{aligned} F_m &= (H_{11} - H_{12}Z_s)(H_{21} - H_{22}Z_s)^{-1}x_m \\ &= \left(\frac{-H_{12}}{H_{21} - H_{22}Z_s} Z_s + \frac{H_{11}}{H_{21} - H_{22}Z_s}\right) x_m = Z_m x_m \qquad (13) \\ Z_m &= \frac{-H_{12}}{H_{21} - H_{22}Z_s} Z_s + \frac{H_{11}}{H_{21} - H_{22}Z_s} \qquad (14) \end{aligned}$$

From the definition of Z_m several observations can be made [9]. In the first place, as it could be expected, the reproduced impedance depends on the dynamics of the H_{ij} terms, that is the bilateral controller structure and tuning. In second place, if we define the **transparency** as the property of exact impedance reconstruction $Z_m \equiv Z_s$, perfect transparency is achieved if all the following conditions are met:

$$H_{11} = 0 \wedge H_{22} = 0 \wedge H_{21} = -H_{12}$$

or equivalently if the matrix H is anti-diagonal. Some comments can be made about the transparency property of the system:

- as the environment impedance decreases ($Z_s \to 0$), the transmitted impedance becomes insensitive to Z_s if $H_{11} \neq 0$, since Z_m depends only on the ratio H_{11}/H_{21};
- as the environment impedance increases ($Z_s \to \infty$), the transmitted impedance becomes H_{12}/H_{22}, which is insensitive to Z_s if $H_{22} \neq 0$.

Rewriting the relationship between master force and position as

$$\begin{aligned} F_m &= (P_r Z_s + P_0)X_m \\ P_r &= \frac{-H_{12}}{H_{21} - H_{22}Z_s} \\ P_o &= \frac{H_{11}}{H_{21} - H_{22}Z_s} \end{aligned}$$

permits to highlight the terms P_r as the **reproducibility** and P_o as the **operationality** [8] properties of the bilateral control system:

- the **reproducibility** is the ability of the bilateral system to faithful reproduce the environment impedance at the master side
- the **operationality** is the property of the bilateral system to not require an additional force at the master side in respect to the force needed at the slave side.

When $P_r = 1$ and $P_o = 0$ the bilateral system impedance replication is transparent to the user and $Z_m \equiv Z_s$ [9]; to obtain the ideal impedance reproduction transparency the following H parameters values are required:

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{15}$$

2.3.3 4 Channels Bilateral Control

The hybrid matrix parameters H_{ij} can be resolved using the 4ch bilateral control schema introduced by [9] and supplemented with the external disturbances compensation by [8] Figure 6, where p indicates high-frequency disturbance that falls outside the disturbance estimation and compensation bandwidth. From the comparison of the 4 channel bilateral control schema and the proposed bilateral controller subsection 2.3.1 we get:

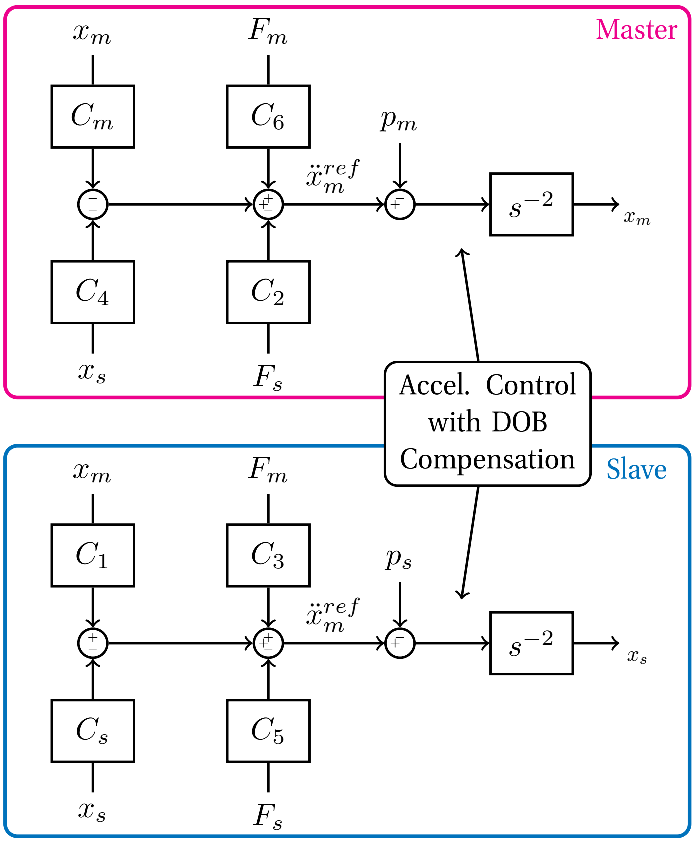


Figure 6: 4ch Bilateral Control diagram, [8]

$$C_1 = -C_4 = C_m = C_s = C_p(s) \tag{16}$$

$$C_2 = C_3 = C_5 = C_6 = C_f(f) \tag{17}$$

The hybrid matrix parameters can now be obtained by resolving the relationships between x_m, x_s, F_m, F_s.

$$H_{11} = \frac{F_m}{x_s} = \frac{C_1C_4 + (s^2 + C_m)(s^2 + C_s)}{C_1C_6 + C_3(s^2 + C_m)} = \frac{s^2}{C_f(s)} \tag{18}$$

$$H_{12} = \frac{F_m}{-F_s} = -\frac{C_1C_2 + C_5(s^2 + C_m)}{C_1C_6 + C_3(s^2 + C_m)} = -1 \tag{19}$$

$$H_{21} = \frac{x_m}{x_s} = -\frac{C_3C_4 + C_6(s^2 + C_s)}{C_1C_6 + C_3(s^2 + C_m)} = 1 \tag{20}$$

$$H_{22} = \frac{x_m}{-F_s} = -\frac{C_2C_3 - C_5C_6}{C_1C_6 + C_3(s^2 + C_m)} = 0 \tag{21}$$

The hybrid matrix H of the considered bilateral controller is then:

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} = \begin{bmatrix} s^2C_f(s)^{-1} & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} s^2/K_f & -1 \\ 1 & 0 \end{bmatrix} \tag{22}$$

from which we can derive that the proposed bilateral control system satisfies the reproducibility condition

$$P_r = \frac{-H_{12}}{H_{21} - H_{22}Z_s} = 1 \tag{23}$$

the environmental impedance is then reproduced at the master side. On the other hand, the operationality is influenced by the design of the force controller, that is the choice of K_f, as reported below:

$$P_o = \frac{H_{11}}{H_{21} - H_{22}Z_s} = \frac{s^2}{Kf} \tag{24}$$

The higher the K_f gain, the lower the operationality P_o value, meaning that the operator needs to apply a smaller additional force to operate the bilateral system in respect to the first-person operation. As it will be derived, the stability of the overall bilateral controller is reduced as the value of K_F increases: the perfect operationality and system stability are conflicting goals.

2.4 EXPERIMENTAL SETUP

2.4.1 Introduction

The experimental setup is composed of two linear motors driven by two servo-drivers, while the control software runs as a real-time kernel module on a general purpose Linux OS computer. Two PCI I/O boards (a D/A converter and a counter board) permit to the real-time control software to read the motors position from the encoders and set the current reference to the servo-drivers. The schematic representation of the experimental setup is presented in Figure 7.

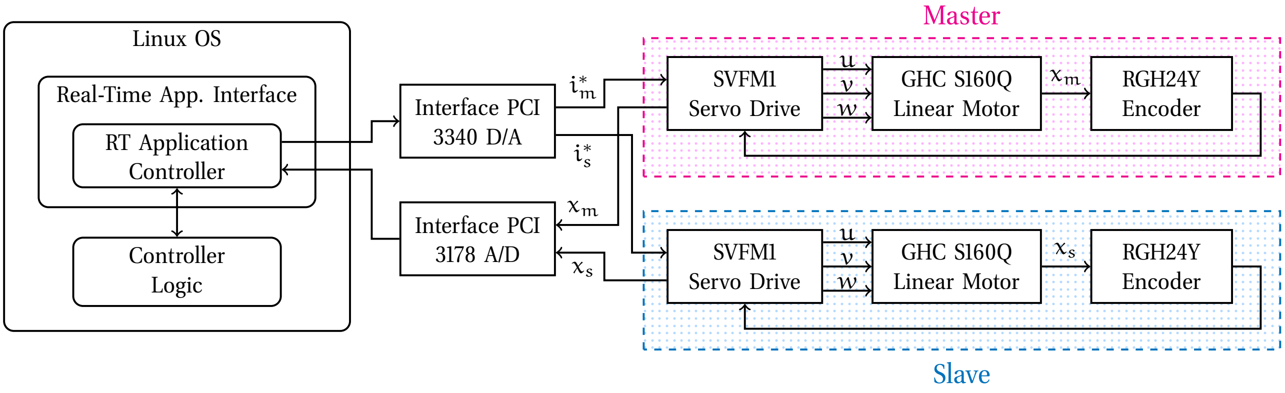


Figure 7: Experimental setup schematic representation.

The use of Permanent Magnet Linear Synchronous Motors (PMLSM) for the generation of the movement offers several advantages over linear motion mechanisms such as ball screws coupled with a rotary motor or rack-and-pinion actuators: in the first place the absence of a mechanical transmission improves the dynamic performances of the motor and removes any kind of backlash, improving the positioning accuracy, an important aspect considering that the system it will be directly handled by an operator. Other advantages include simpler construction, high force density and higher reliability because the only parts in contact and subject to wear are the linear bearings. The disadvantages of using PM linear synchronous motors are the high cost of the motor due to the large amount of rare-earth permanent magnet material used for the PM mover construction and the pronounced force ripple. The force ripple is caused by both the *end effect* and the *cogging torque*. The cogging torque is caused by the magnetic interaction between the stator slots and the permanent magnets; the end effect is caused by the discontinuity of the magnetic circuit near both ends of the motor that decreases the thrust, increasing power consumption and degrading the motor performances.

2.4.2 Real-Time Controller

A computer running CentOS Linux is used as hardware-in-the-loop for rapid prototyping of the system controller and for the data saving and post processing. The control software is written in the C programming language using the Real-Time Application Interface library (RTAI) to observe the control timing constraints: the RTAI library permits to run the controller application as a Real-Time kernel module. Data buffers are used to share data between the RT and non-RT processes, like loading motor position reference and saving the motor speed and current; the data post-processing is done primarily using Python scripts and the the NumPy library. The RTAI real-time kernel scheduler has been configured to achieve a control frequency of $f = 10\,\text{kHz}$. In order to guarantee the timing constrains set by the real-time Linux subsystem scheduler, the real-time application cannot perform disk access or other "slow" I/O operations, except for the PCI bus read/write. Dedicated RTAI buffers, similar to UNIX `pipes`, are set up for the communication between the RT controller and the non-RT application logic.

2.4.3 Linear Motor and Position Encoder

A photo of the experimental setup linear motor is shown in Figure 8: the motor body is fixed in place while the PM mover can slide back and forth thanks to the linear ball slides, model THK LS 1077. The linear encoder is integral with the aluminum body that holds all the component in place. The exact same setup is duplicated creating a matching master/slave systems pair. The model of the linear motors is Nippon Pulse S160Q and the main parameters are reported from the datasheet in Table 1. The motor travel is limited by the travel of the linear bearing, that is $100\,\text{mm}$.

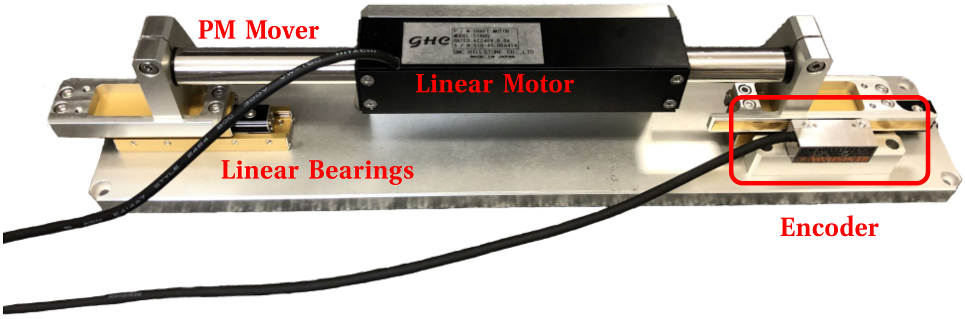


Figure 8: Experimental setup: linear motor and position encoder.

Table 1: Nippon Pulse S160Q Linear Motor main parameters.

Specification	Value	
Continuous Force	20	N
Continuous Current	0.62	A
Max Force	81	N
Max Current	2.5	A
Force Constant K_f	33.0	N/A
Windings Resistance	43	Ω
Windings Inductance	16	mH
Motor Travel	163	mm

POSITION ENCODER The motor positions are obtained by two Renishaw RGH24Y15A30A incremental encoders: they use a highly reflective gold plated tape (sticked to the linear motor mover) to measure the motor position contactless. The output of each encoder is composed by two digital signals with a 50% duty-cycle, phase shifted by 90° and with a period of 4 µm: by checking the logic level of the two signals and counting the total amount of steps since the controller start-up, the motor position can be obtained with a resolution of 1 µm. The encoders are not absolute, so the motors need to be homed after every controller reset; the homing procedure is done manually by zeroing the encoder count when the motor are positioned to the mechanical stop. The main characteristics of the encoders are reported in Table 2.

Table 2: Renishaw RGH24Y15A30A position encoder specification.

Specification	Value	
Supply Voltage	5	V
Max Current Consumption	120	mA
Resolution	1	µm
Max Speed	8	m/s

2.4.4 Servo-Driver

Each linear motor is driven by a servo-driver SVFM1 produced by Servoland, with main characteristics reported in Table 3. The motor current reference for the servo-driver is an analog signal generated by the PCI DAC board. The driver apply the proper voltage to the motor windings to mantain the current proportional to the analog reference. The maximum current output of the servo-driver is 2 A, which is lower than the motor maximum current rating (2.5 A) so no additional limitations have been set on the servo driver. The servo-drivers act like pass-through for the encoder signals, which are then sent to the counter board and back to the PC through the PCI bus.

Table 3: Servoland SVFM1 servo-driver main characteristics.

Specification	Value	
Supply Voltage	110	V
Max Output Voltage	84	V
Max Output Current	2.0	A
Transconductance Gain	0.2	A/V
Reference Range	±10	V

2.4.5 Multifunction PCI I/O board

The I/O boards function as the interface between the two servo-drivers and the computer, enabling the real-time control program to set the servo-drivers

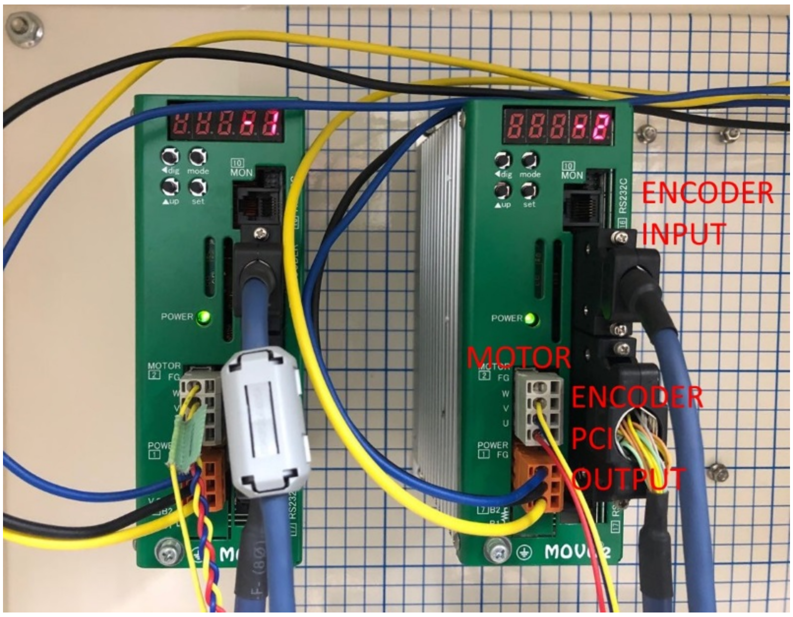


Figure 9: SVFM1 Servo Driver

current reference and to read the motors position from the encoders. Their main characteristics are presented in Table 4.

Table 4: Multifunction PCI I/O boards specifications

Board	Type	Specification	Value	
PCI-6201	Counter Board			
		Input Range	[0; 5]	V
		Channels	4	
		Resolution	24	bit
		Max Input Freq.	2	MHz
PCI-3340	DAC			
		Output Range	[−10; 10]	V
		Channels	8	
		Resolution	16	bit
		Conversion Time	20	µs

D/A Converter

Given that the output range of the DAC board matches the current reference input range of the servo driver (±10V), the voltage needed to set the servo-driver current reference is simply the current reference divided by the servo transconductance gain:

$$u^{ref}_{cur} = K_v \, i^{ref} = \frac{i^{ref}}{0.2} = 5 \cdot \, i^{ref} \tag{25}$$

Counter Board

As presented earlier, each encoder output consists of two digital signals that toggle their logic level every time the motor travels $2\,\mu m$ and, because the two signals are phase-shifted by 90°, the movement direction is also known. To translate the logic signals from the encoders into the motors positions a counter board is used, specifically the PCI 6201 made by *Interface*. The maximum input signal commutation frequency of the counter board is 2 MHz, that translates into a maximum velocity of

$$v_{max} = 2\,\mu m \cdot 2\,MHz = 4\,m/s \tag{26}$$

above which the board is going to miss some encoder steps, causing a position miscalculation. Another characteristic of the counter board that could interfere with the system functioning is the number of counter bits (23 plus the sign bit), which limits the maximum measurable distance before the counter overflows to:

$$d_{max} = 2^{23} \cdot 1\,\mu m = \pm 8.4\ m \tag{27}$$

Neither of these two limitations impact the experimental setup, where the motor travel is limited to 10 cm and, with the hypothesis that the master system is controlled by hand, the speeds involved are significantly lower than v_{max}. This last assumption is supported by [12], where it is presented that for the tested subjects ($N = 40$) the hand movement peak-velocity never exceeds 3 m/s and it is usually around 2 m/s for young healthy adults. The maximum observed velocity in the tests performed doesn't however exceed 0.3 m/s. If any of these conditions should not meet new testing criteria the encoders precision can be lowered, increasing the maximum speed v_{max} and the maximum distance d_{max} before the counter board overflow.

2.5 MECHANICAL SYSTEM MODEL

The first step for designing the control system is to have a mechanical system model. A simplified model of the system is presented in Figure 10.

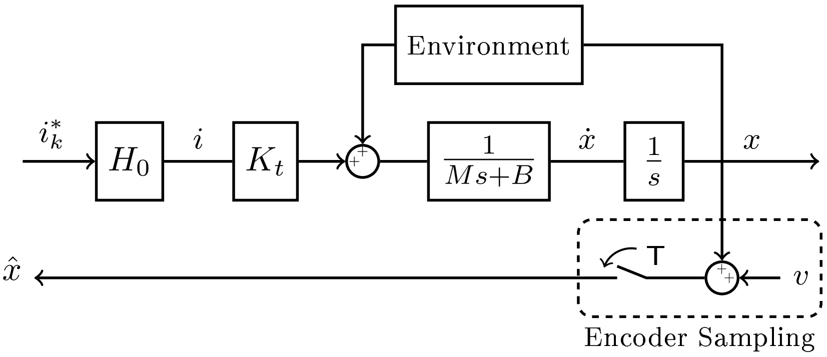


Figure 10: Motor System Model

From the point of view of the bilateral controller, the only controllable motor input is the current reference i_k^* that is converted by the DAC board to the voltage u_{cur}^{ref} (25) and sent to the servo-driver. From the controller standpoint, the D/A converter acts like a zero-order holder H_0. The servo-driver then applies an input voltage to the motor windings in order to generate the reference current. From previous motor usage, an used derivative (speed) gain is $K_v^\dagger = 0.7\,\mathrm{As/m}$: the motor mechanical time constant is then

$$\tau_m = \frac{M_n}{B + K_v^\dagger K_t} \simeq \frac{M_n}{K_v^\dagger K_t} \simeq 26\,\mathrm{ms} \tag{28}$$

Given that the linear motor electrical time constant

$$\tau_{el} = \frac{L}{R} = \frac{16\,\mathrm{mH}}{43\,\Omega} = 0.372\,\mathrm{ms} \tag{29}$$

is two orders of magnitude smaller than the mechanical one, the servo-driver current/force dynamics is neglected in the design of the controller. Instead, the static relation

$$F_k \simeq K_t \cdot i_k \simeq K_t \cdot i_k^* \tag{30}$$

is used, where i_k^* is the current reference at the k-th step. The servo driver is then modeled as a zero-order holder with a time-delay equal to the sampling time T:

$$H_0 = \frac{1 - e^{-sT}}{sT} \tag{31}$$

The actual motor is modeled as a first-order system with mass M and viscous friction B, ignoring for the moment non-idealities such as the static friction.

Position Encoder – Controller Communication

The PCI counter board count each step that the Renishaw encoders register. The result, multiplied by the encoder resolution q, represent the quantized motor position relative to the the starting point, that is the motor position at the system startup.

2.6 MECHANICAL SYSTEM IDENTIFICATION

In this section is described the procedure for the identification of the main mechanical system parameters: motors moving mass, static and Coulomb friction, thrust constant.

2.6.1 Motor Mass and Force Constant

The motor mass is measured by vertically aligning the motor axis over a scale and weighting the mover mass. The nominal mass for both linear motors is 0.6 kg, while from the measurements:

$$\begin{aligned} m_1 &= 0.575 \pm 0.005 \text{ kg} \\ m_2 &= 0.580 \pm 0.005 \text{ kg} \end{aligned}$$

With the motor still in the vertical configuration and now knowing the mover mass it is possible to estimate the actual force constant K_t by measuring the current needed to balance the weight of the motor. For this test a simple proportional position controller is used with $K_p = 1$ while the position reference is varied along the motor travel. The steady-state currents i_{ss} needed for balancing the weight of the motor mover for $N = 8$ different motor positions have been recorded. All the samples fall into the following range

$$i_{ss} \in [0.162;\ 0.199]\ \text{A} \tag{32}$$

From the steady state current references the force–current relation can be obtained as

$$\hat{K}_t = \frac{mg}{i_{ss}} = \frac{0.58 \cdot 9.81}{i_{ss}} = 31.8 \pm 3.3 \text{ N/A} \tag{33}$$

The uncertainty range of the $\hat{K}_t$ estimation includes the nominal value $K_{t,n} = 33\,\text{N/A}$, with the maximum relative error of 10.3%. The nominal value (33) is greater than the estimates average (31.8) by about the 3%: this could be explained by the low samples count or by the inevitable presence of mechanical friction that helps to hold the motor (even if the motor weight force is not acting on the linear bearings). The force constant fluctuations are caused by the permanent magnet position in respect to the stator windings, that varies with the mover position, and by the cogging torque that affects the PM linear motors.

2.6.2 Friction Estimation

The (crude) estimation of the mechanical system friction is necessary to determine the environment reaction force for the bilateral force control from the total estimated external disturbance. For the estimation of the friction a constant speed trajectory is used as the motor reference while recording the motor current: because no external forces are applied to the motor and the inertia term is null ($\ddot{x}_m = 0$), the force and thus the current necessary to follow the reference is solely imputable to the mechanical friction. Given the linear motor limited travel (about 10 cm) the position reference used is a symmetric trapezoidal trajectory: by changing the movement duration while keeping the travel length constant is possible to obtain runs at different speeds. The friction is modeled as the sum of the static friction term F_{st} and the Coulomb friction B:

$$F = F_{st}\,\mathrm{sign}(\dot{x}) + B\,\dot{x} \tag{34}$$

An example of the position and speed references for $v^{ref} = \pm 6\,\mathrm{cm/s}$ is presented in Figure 11 as well as the friction force calculated as the nominal force constant $K_{t,n}$ multiplied by the current reference. For each velocity run the friction force is calculated as the mean of the force during the movement (orange line) only when the velocity is constant (the error is within 5% of the reference value, light red background).

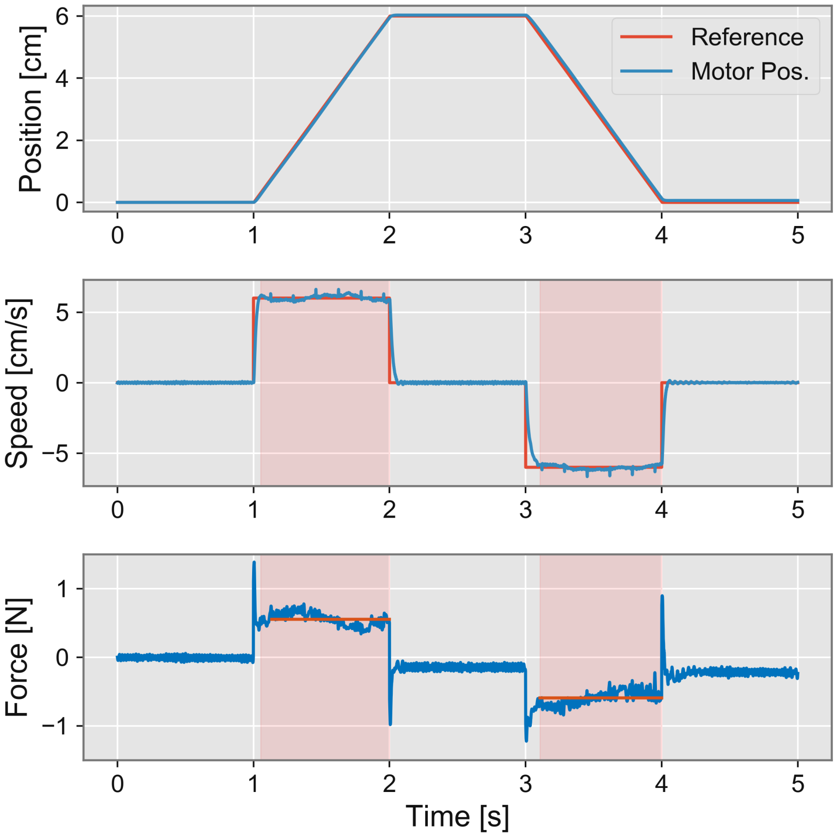


Figure 11: Fricton estimation – Position error, speed and force (current) during a test run with $v^{ref} = \pm 30\,\mathrm{mm/s}$. For the friction estimation only the samples withing 5% of the steady-state speed are considered (red background).

FIRST MECHANICAL SYSTEM The test procedure is executed for different speed references: the collected speed–friction force data are summarized in Figure 13b. From the linear interpolation of the collected samples:

$$
\begin{aligned}
B^{+} &= 3.67 \text{ Ns/m} \\
B^{-} &= 3.82 \text{ Ns/m} \\
F_{st}^{+} &= 0.32 \text{ N} \\
F_{st}^{-} &= 0.21 \text{ N}
\end{aligned}
$$

where the suffix $\bigcirc^{+}$ refers to positive speeds and $\bigcirc^{-}$ to negative ones.

SECOND MECHANICAL SYSTEM The process is repeated for the second mechanical system (*slave*). The results are reported in Figure 13a.

FRICTION RESULTS The results for the two systems and the two conditions are synthesized in Table 5. Given the similar results between the conditions a common model is proposed with average values for the viscous (3.7 Ns/m) and static (0.25 N) friction.

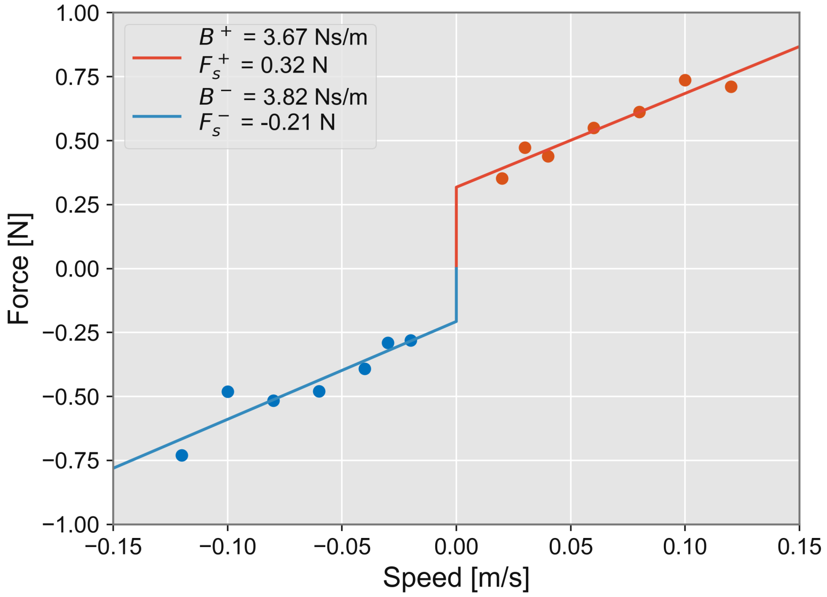
Table 5: Friction estimation data

System	Conditions	Viscous Friction [Ns/m]	Static Friction [Ns/m]
Master	Positive speed	3.67	0.32
Master	Negative speed	3.82	0.21
Slave	Positive speed	3.77	0.24
Slave	Negative speed	3.69	0.25
	Average	3.73	0.255

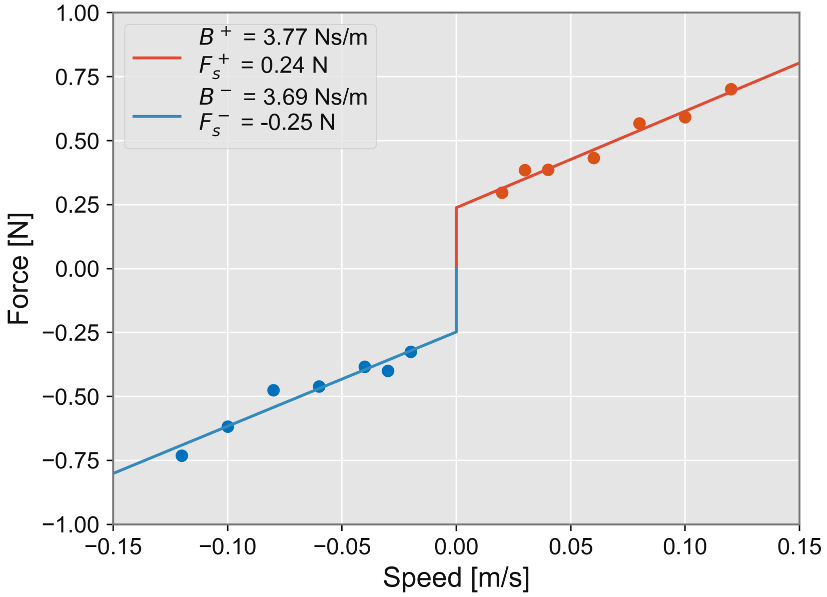
2.7 THE DISTURBANCE OBSERVER

What characterize a bilateral control system from other control strategies is the shared force feedback between the master system and the slave, that provides improved tactile sensibility to the operator: it is therefore necessary to measure or estimate the force acting on each motor to mirror it at the other end of the bilateral system. The use of force sensors to measure the external forces applied to the robots constitutes an additional component added to the total cost of the system and its overall reliability. Among all the available technologies for the realization of force sensors, piezoelectric load sensors have sufficiently high bandwidths (> 100Hz) for the bilateral control readiness, but, besides being expensive, they are prone to substantial measurement drift under light loads, requiring some sort of calibration procedure. Instead of using an actual load sensor, the bilateral control system makes use of the Disturbance Observer (DOB). A disturbance observer employs the motor model in conjunction with the current reference and the motor velocity to estimate the total external disturbance acting on the motor, disturbance that could be introduced by the operator, the reaction force of the environment or other motor non-idealities not present in the model.

Figure 12: Friction estimation data interpolation – Positive (red) and negative (blue) velocities.



(a) Master mechanical system



(b) Slave mechanical system

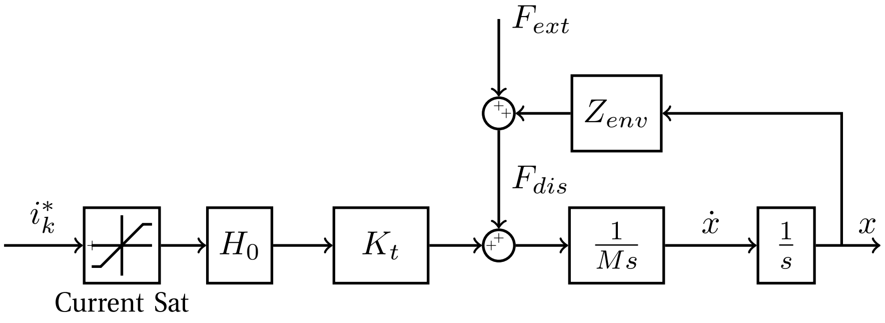


Figure 14: External disturbances acting on the motor.

In Figure 14 the external disturbance acting on the motor is illustrated: F_{dis} can be estimated by subtracting the nominal inertia term $M_n\ddot{x}$ from the motor thrust, calculated from the current reference i^* and the nominal force-current constant $K_{t,n}$:

$$\hat{F}^*_{dis} = K_{t,n}i^* - M_n\ddot{x} \tag{35}$$

The acceleration signal is obtained by deriving twice the position signal: the derivation of a quantized signal introduces noise that corrupts the acceleration estimate and so the external force estimate. Because the estimated disturbance is used later for the compensation of the non-idealities and by the force controller, the noisy signal enters the control loop. Therefore the quantization noise could generate the actuator *ringing* caused by the high-frequency control noise: to cope with this problem a low-pass filter with bandwidth g_{dob} is introduced:

$$\hat{F}_{dis} = \frac{g_{dob}}{g_{dob}+s}\hat{F}^*_{dis} \tag{36}$$

It is possible to avoid the derivation of the speed signal rearranging the operations:

$$\begin{aligned}
\hat{F}_{dis} &= \frac{g_{dob}}{g_{dob}+s}\left[K_{t,n}i^* - M_n s^2 x\right] \\
&= \frac{g_{dob}}{g_{dob}+s}\left[K_{t,n}i^* - M_n s^2 x \pm g_{dob}M_n s x\right] \\
&= \frac{g_{dob}}{g_{dob}+s}\left[K_{t,n}i^* + g_{dob}M_n s x - M_n s x\,(s+g_{dob})\right] \\
&= \frac{g_{dob}}{g_{dob}+s}\left[K_{t,n}i^* + g_{dob}M_n\dot{x}\right] - g_{dob}M_n\dot{x}
\end{aligned}$$

The disturbance estimate is used by the force controller to implement the haptic feedback between the master and the slave systems, but also for the feed-forward compensation of the external disturbances. Thanks to the DOB, from the perspective of the controller the motor perform as a nominal one, with mass M_n and force/current constant $K_{t,n}$, at least those with frequencies sufficiently lower than the DOB bandwidth. The presence of the DOB justifies the absence of an integral action in the position controller, because the constant or slowly-varying disturbances are estimated and compensated for. In Figure 15 is presented the DOB block schema for a single motor/DOF. The nominal motor model is composed by a constant force/current relationship, while the viscous friction is included into the environment impedance

Z_{env}; other external disturbances, such as static friction or mechanical non-idealities, are included into F_{ext}

$$F_{dis} = Z_{env}\, x + F_{ext}$$

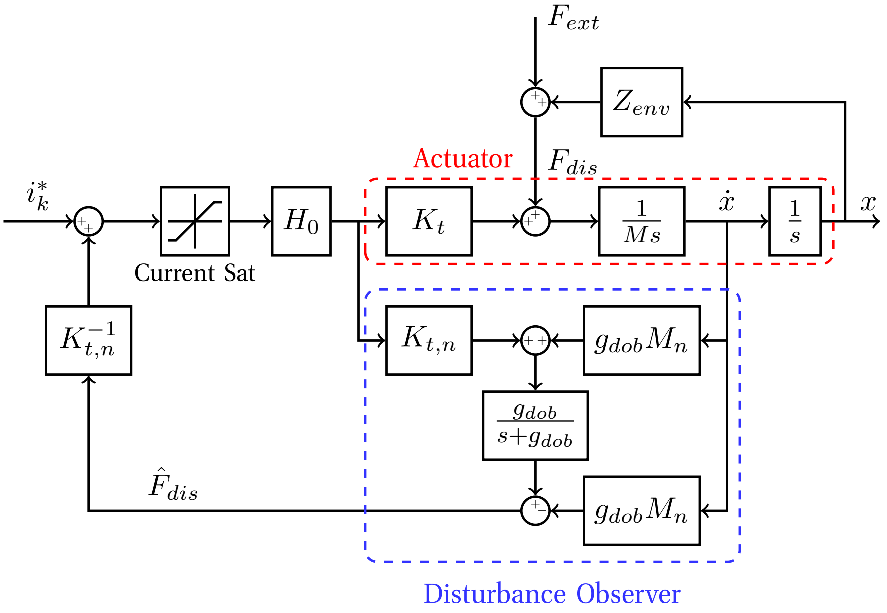


Figure 15: Disturbance Observer control schema – compensation of system non-idealities

Current Reference Saturation

An important control design aspect is the presence of a current reference saturation block in the control scheme: as we saw in Table 3 the servo-driver has a maximum current limit of 2 A, that will cap the current reference whenever it exceeds this value. Other that a software fail-safe, the saturation block ensures that the DOB input is the same as the actual motor, otherwise in presence of a current saturation by the servo-driver the estimated external force could diverge, because the observer input differs from the actual one.

2.7.1 Model Non-Idealities Compensation

By supplementing the controller current reference i^* with the current associated to the estimated disturbance i_{dis},

$$i_{dis} = \frac{\hat{F}_{dis}}{K_{t,n}} \tag{37}$$

from perspective of the bilateral controller the mechanical system performs as the nominal one ($K_{t,n}$, M_n, null friction). In fact, with the hypothesis that the DOB bandwidth is larger than the disturbances', the estimated disturbance value is

$$\hat{F}_{dis} = \left[F_{dis} + (M - M_n)s^2 x + (K_{t,n} - K_t)\, i\right] \frac{g_{dob}}{s + g_{dob}} \tag{38}$$

that is the external disturbance, the errors between actual and nominal mass and force constant. Whereas the mechanical system is characterized by

$$x = \frac{1}{Ms^2}\left[K_t\left(i + \frac{\hat{F}_{dis}}{K_{t,n}}\right) - F_{dis}\right] \tag{39}$$

after the simplifications, the compensation of the model non-idealities yields

$$x = \frac{1}{M_n s^2}\left[K_{t,n}\; i + \hat{F}_{dis}\frac{s}{s + g_{dob}}\right] \tag{40}$$

The mechanical system now perform like a nominal one, with mass M_n, no friction and nominal force/current constant, with the addition of the high-frequencies components of F_{dis} that fall outside of the DOB estimation bandwidth.

2.7.2 Speed derivative

In the DOB formulation (Figure 15) it is assumed that the motor speed $\dot{x}$ is directly measurable. The motor velocity is actually obtained by deriving the position that, as saw in section 2.5, is corrupted by the quantization noise. To attenuate the high frequency noise the velocity derivative is filtered by a first-order low-pass filter with bandwidth g_s, but it introduces a phase delay. In this particular setup, the fine position encoder precision and high sampling frequency permit to keep the filter bandwidth sufficiently high. Following the same rationale used in the disturbance observer derivation, the speed $\dot{x}$ of each linear motor is calculated as a filtered derivative of the position:

$$\dot{x} = \frac{s g_s}{g_s + s} x \tag{41}$$

to avoid the direct derivation of the position signal, it is implemented as

$$\hat{\dot{x}} = \frac{g_s}{g_s/s + 1} x$$

which represent a numerically more robust solution.

2.7.3 Software Implementation

The described filtered speed derivative and disturbance observer are implemented in C as follows.

Listing 1: C implementation of the velocity and DOB estimators.

```
/* Motor nominal mass [kg] */
#define M       (0.6)
/* DOB LPF freq. [rad/s] */
#define G_DOB   (600)
/* Speed derivative LPF freq. [rad/s] */
#define G_S     (600)
/* Nominal force constant [N/A] */
#define KT      (33)

/* Motor structure */
struct joint {
    double x;       /* position */
    double dx;      /* velocity */
    double f_dob;   /* estimated disturbance force */
    double f_cmd;   /* force command (= i*Kt) */

    /* var for velocity integration */
    double int_dx;
    /* var for dob integration */
    double int_dob;
};
struct joint jm, js;

/* velocity estimation */
void velocity_update(struct joint *j) {
        j->int_dx += j->dx*T;
        j->dx = (j->x - j->int_dx)*G_S;
}

/* disturbance observer */
void dob_update(struct joint *j) {
    j->f_dob = j->int_dob - j->dx*M*G_DOB;
    j->int_dob +=
        ((j->f_cmd + j->dx*M*G_DOB) - j->int_dob) * G_DIS*T;
}

/* [...] in the control loop */
    jm.x = read_pos(chm);
    velocity_update(&jm);
    dob_update(&jm);

    /* Control logic [...] */

    /* Write current reference to DAC -> servo */
    write_out(jm);
```

2.7.4 DOB bandwidth

The choice of the speed pseudo-derivative and the DOB bandwidths g_s, g_{dob} is not straightforward:

- higher bandwidths have faster dynamic responses, but the output is more corrupted by the high-frequency noise
- lower bandwidths produce cleaner output signals, but the limited bandwidth attenuates both the noise and the high frequency transients

In Figure 16 is presented the speed estimations of the same motor trajectory with different bandwidths g_s during a constant speed ramp. For this time-limited example the data is saved at the controller sampling frequency $T = 0.1\,\mathrm{ms}$ to better study the DOB performance. As g_s decreases:

- the estimated speed overshoot is damped and appears later in time
- the signal is less corrupted by the quantization noise

Because fast speed and disturbance estimations are important for a responsive bilateral control system, it is chosen a bandwidth of $600\,\mathrm{rad/s}$ for both, which trades some noise for faster dynamics.

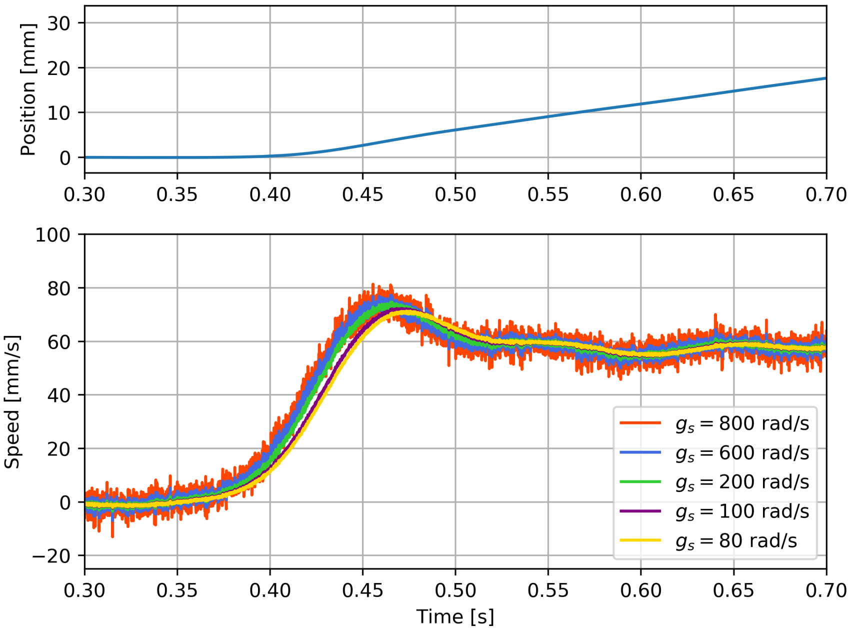


Figure 16: Speed estimation using different pseudo-derivative bandwidths.

2.7.5 Disturbance Observer Robustness

The DOB robustness can be assessed by studying the system transfer function with non nominal motor parameters. Using Figure 15, the relationship between motor speed and current is

$$\frac{\dot{x}(s)}{i^*(s)} = \frac{K_{t,n}(s+g_{dob})}{M_n s\left(\frac{MK_{tn}}{M_nK_t}s + g + \frac{BK_{tn}}{M_nK_t}\right)} = \frac{K_{t,n}(s+g_{dob})}{M_n s\,(\alpha s + g')} \tag{42}$$

that is the nominal model $K_{t,n}/M_n$ multiplied by a zero and pole pair. Said pair behaves like a phase lead network for

$$\alpha = \frac{MK_{tn}}{M_n K_t} \leqslant 1 \quad \wedge \quad B \geqslant 0$$

If these conditions are not respected, the zero-pole pair introduces a phase delay that decreases the outer control loop stability margin: a condition we would like to avoid. While the viscous friction B is always greater than zero, the other condition is guaranteed by using, for the DOB design, the nominal mass $0.6\,\mathrm{kg}$ (while the actual mass is $0.58\,\mathrm{kg}$) and the nominal motor force/current constant $33\,\mathrm{N/A}$. For pure fortuity:

$$\alpha = \frac{0.58\,\mathrm{kg}}{0.6\,\mathrm{kg}} \cdot \frac{33\,\mathrm{N/A}}{31.8\,\mathrm{N/A}} = 1.003$$

The nominal DOB parameters were not changed.

2.8 BILATERAL CONTROLLER IMPLEMENTATION

In Figure 17 the overall control block diagram for the Bilateral Control system is presented. The position controller output (the acceleration reference $\ddot{x}_c$) and the force controller output ($\ddot{x}_d$) are combined to obtain the acceleration references for the master and slave systems ($\ddot{x}_m, \ddot{x}_s$) that are converter to current references with the nominal mass M_n and force constant $K_{t,n}$. The current references are then supplemented with the estimated disturbances compensation current before the current saturation block (limit set to $2\,A$). The current reference is then applied to the linear motor by the servo driver, which dynamics is ignored in the block diagram, producing a force that moves the motor. The motor movement is read by the position encoder and it is used to estimate the motor speed. The speed, combined with the current reference, is then used to estimate the motor equivalent external force F^{dist}. The master and slave positions, speeds and external forces are the inputs for the bilateral controller.

2.8.1 Communication Delay

From the perspective of the control signals, the bilateral controller (the hardware-in-the-loop industrial computer used in this work) receives the position signals from the master and slave mechanical system encoders and generate the control current references for the servo-drivers. The described setup communication delay between the 3 components is negligible compared to the control period T because the servo-drivers and the encoders (for which the servo-driver acts like a pass-through) communicate directly with the bilateral controller through the high-speed PCI bus. In other contexts where the communication delay isn't negligible the robustness of the system must be asserted. Several solution has been tested to compensate for the communication delay, for example the use of a CDOB [6], a Communication Disturbance Observer, that treats the communication delay as a disturbance interfering with the system, that is estimated by the CDOB and compensated by a feedforward action.

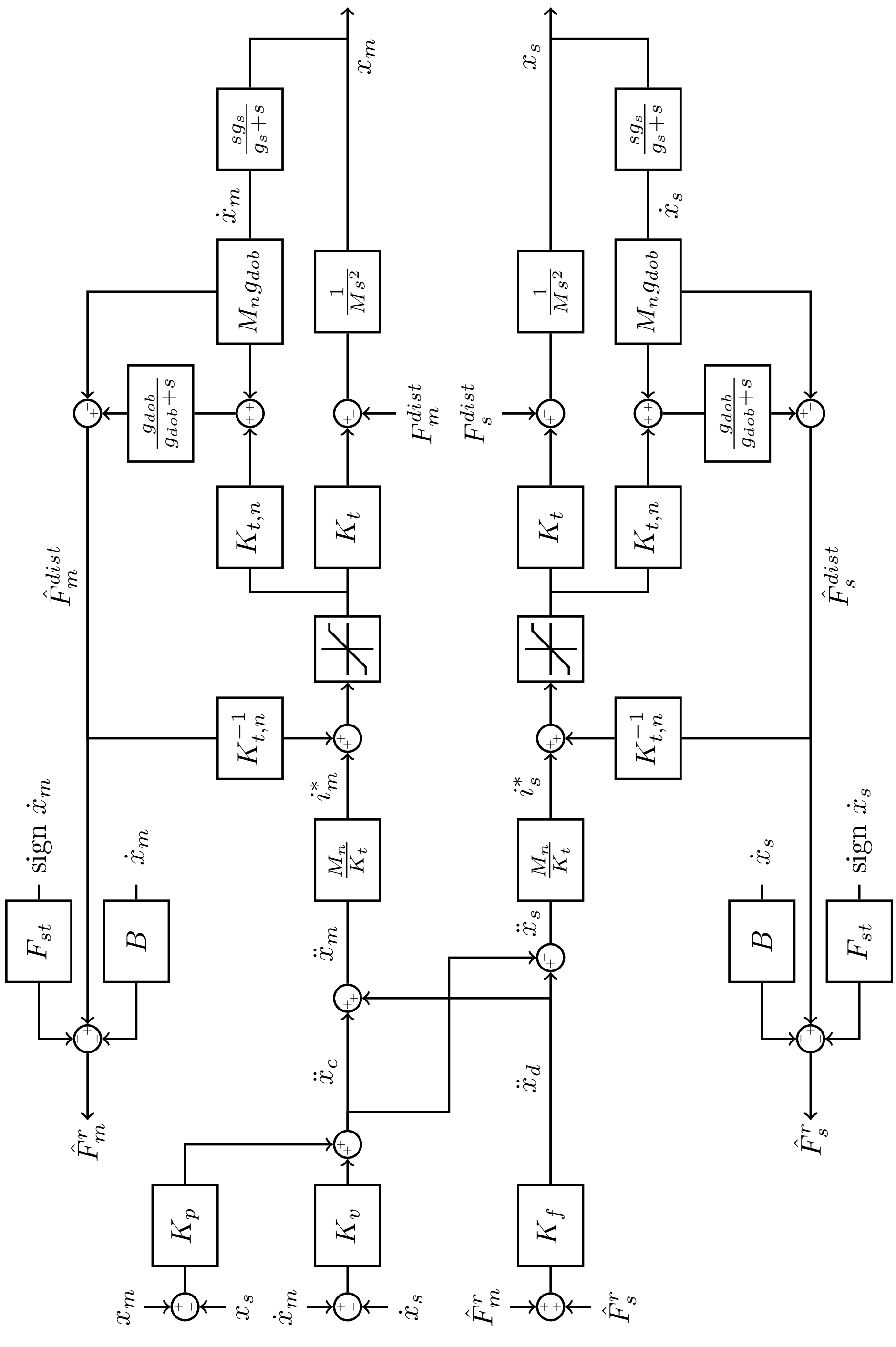


Figure 17: Bilateral Control block diagram

2.9 BILATERAL CONTROLLER DESIGN

At this point the mechanical system model main parameters are known and the external disturbances are estimated by the disturbance observer. As presented before, thanks to the modal decomposition, the design of the bilateral controller can be split into the design of the position controller (differential mode) and the force controller (common mode).

2.9.1 Position Controller Design

Before proceeding with the position controller design, it is appropriate to consider how it's integrated into the bilateral control. The first consideration is that the motor acceleration and current reference can be used interchangeably as they are directly related by the nominal mass and force constant:

$$i^* = \frac{M_n}{K_{t,n}} \ddot{x}^*$$

The position controller inputs are the position and speed errors between master and slave; the output is the current/acceleration reference that will be added to (subtracted to) the force controller output to drive the master side (slave side). This way both mechanical systems contribute to the reduction of the position and speed errors by moving towards each other. Therefore the position controller can be designed considering only one mechanical system, given a position/speed reference.

Position LQR Controller

The design of the position controller is based on the Optimal Linear Quadratic Regulator theory: from the state-space model of the mechanical system and a quadratic cost function, the synthesized controller minimizes the specified control cost during a finite or infinite time horizon. The space-state model is derived from the mechanical model presented in Figure 10: the state variables are the motor position x and speed $\dot{x}$, the only controllable system input is the motor current i^* and the only measurable output the motor position:

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \\ \dot{\mathbf{x}}(t) &= F\,\mathbf{x}(t) + G\,u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(t) = x(t) \end{aligned}$$

with

$$F = \begin{bmatrix} 0 & 1 \\ 0 & -B/M_n \end{bmatrix} \qquad G = \begin{bmatrix} 0 \\ K_{t,n}/M_n \end{bmatrix}$$

With the hypothesis that the external disturbances like static and viscous friction, non-constant force/current relation K_t, non-nominal mass $M \neq M_n$ and other mechanical non-idealities are compensated by the DOB, it is set $B = 0$ for the design of the position control. Because the controller operates between discrete time intervals, the first step is to obtain the equivalent state-space discrete-time system. From the Systems Theory, the state of a dynamic system evolves from the instant t_k like

$$\mathbf{x}(t) = e^{F(t-t_k)}\mathbf{x}(t_k) + \int_{t_k}^{t} e^{F(t-\tau)} G u(\tau)\, d\tau \tag{43}$$

After the time $T = t_{k+1} - t_k$, equal to the sampling time, during which it is supposed that the input is kept constant by the DAC holder, the state evolves to

$$\mathbf{x}(t_{k+1}) = e^{FT}\mathbf{x}(t_k) + \int_{0}^{T} e^{F\tau} G\, d\tau\, u(\tau) \tag{44}$$

Finally, the equivalent discrete model looked for:

$$\mathbf{x}[k+1] = \Phi\ \mathbf{x}[k] + \Gamma\ u[k] \quad \text{and} \quad \Phi = e^{FT},\ \Gamma = \int_{0}^{T} e^{F\tau} G\, d\tau \tag{45}$$

where $\Phi \in \mathbb{R}^{n\times n}$ and $\Gamma \in \mathbb{R}^{n\times m}$. If we consider an infinite time horizon, the control goal of the linear quadratic controller is to find the input sequence $u(\cdot)$ that minimize the quadratic loss function

$$J(\mathbf{x}_0, u) = \sum_{k=0}^{\infty} \mathbf{x}^T[k] Q \mathbf{x}[k] + u^T[k] R u[k] \tag{46}$$

where $Q \in \mathbb{R}^{n\times n}$ (weights on the state) is a symmetric and semi-definite positive square matrix and $R \in \mathbb{R}^{m\times m}$ (weights on the input usage) a square, symmetric and definite positive matrix. From the linear quadratic optimal control on infinite time horizon theory, given that Q is positive semi-definite and so it can always be decomposed into $Q = C^T C$, and M is the symmetric semi-definite positive solution of the *Algebraic Riccati Equation* (**ARE** (47))

$$M = Q + \Phi^T M \Phi - \Phi^T M \Gamma \left[R + \Gamma^T M \Gamma\right]^{-1} \Gamma^T M \Phi \tag{47}$$

the optimal control input that minimizes the loss index $J(\mathbf{x}_0, u)$ is

$$u[k] = -\left[R + \Gamma^T M \Gamma\right]^{-1} \Gamma^T M \Phi\ \mathbf{x}[k] = -K\ \mathbf{x}[k] \tag{48}$$

From the LQR theory said state feedback is stable i.f.f.

$$\begin{bmatrix} zI - \Phi \\ C \end{bmatrix} \text{ has full rank } \forall\ z \in \mathbb{C},\ |z| \geqslant 1 \tag{49}$$

From the (48) the system optimal input is a time-invariant state feedback, calculable in MATLAB as

$$K = \mathrm{dlqr}(\Phi,\ \Gamma,\ Q,\ R)$$

LQR Weights Choice

After entering the motor mass (0.58 kg), force nominal constant (33 N/A) and $B = 0\,\mathrm{Ns/m}$, the equivalent discrete matrices of the test bench mechanical system is

$$\Phi = \begin{bmatrix} 1 & 10^{-4} \\ 0 & 1 \end{bmatrix} \qquad \Gamma = \begin{bmatrix} 2.85 \cdot 10^{-7} \\ 5.69 \cdot 10^{-3} \end{bmatrix} \tag{50}$$

We can partition the weight matrices as follows:

$$Q = \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix}, \quad q_1,\ q_2 \in \mathbb{R}$$

and from the condition (49) we see that $q_1 \neq 0$, otherwise the rank of (49) drop to 1 for $z = 1$; the other condition is that R is definite positive that, because R is has dimension 1, translate to $R > 0$.

2.9.2 Simulation Tuning and Results

The tuning of the position LQ controller (that is the choice of $q_1,\ q_2$ and R) is done initially in simulation, using the discrete-time motor model $(\Phi,\ \Gamma)$, by evaluating the system response to a step-variation of the position reference. The controller tuning starts from the initial values $q_1 = q_2 = R = 1$, Test A. The system response in Figure 18 shows a non-satisfactory dynamics, with the $1\,\mathrm{cm}$ step response rise time of more than $2\,\mathrm{s}$. Two factors limit the system dynamics: the R weight on the current reference (limiting the amount of current used) and the Q_2 weight on the speed error (limiting the maximum moving speed). The first corrective action taken was the lightening of the input (current) weight R in Test B ($R = 0.1$) and Test C ($R = 0.01$). The system rise time, overshoot and eigenvalues $\lambda_{1,2} = \text{eig}\left[F - GK\right]$ are reported in Table 6; the improvements were negligible, reducing the rise time by a mere $4\,\mathrm{ms}$, as confirmed by the unchanged slower eigenvalue 0.9990.

The next line of action is to change the weights on the state, $q_1,\ q_2$. First, the position error weight is increased to $q_1 = 10$, Test D: the resulting controller improves the system dynamics, reducing the rise-time more than 3 times at $0.7\,\mathrm{s}$. Further improvements are achieved lightening the speed weight q_2:

- with $q_2 = 0.1$ the rise-time is now $220\,\mathrm{ms}$;
- to guarantee the control stability the (49) imposes $q_1 \neq 0$, but no conditions are given for q_2, that is now set to zero. With $q_2 = 0$ the rise-time is further reduced to $52\,\mathrm{ms}$, but the system eigenvalues are now complex conjugates: the step response generates an overshoot of more than 4%;
- with $q_2 = 0.01$ the system eigenvalues are still complex conjugates, but the overshoot is now smaller than 0.1%;
- with $q_2 = 0.02$ the system eigenvalues are purely real, there is no overshoot and the rise-time is $0.101\,\mathrm{s}$;

The system step responses are reported in Figure 19. The best compromise is offered by Test G and Test H controllers, that achieve a fast step response with a null or negligible overshoot.

From these tests the speed weight q_2 appears to be the main discriminant regulating the step response behavior. The final choice is postponed until the experimental results confirm the behavior seen in the simulation.

Table 6: LQR Position Controller – MATLAB Simulation

Test	Q_1	Q_2	R	K	$\lambda_{1,2}$	T_r [s]	Overshoot
A	1	1	1	[0.976 0.888]	[0.9990 0.9470]	2.211	-
B	1	1	0.1	[2.911 2.819]	[0.9990 0.8428]	2.198	-
C	1	1	0.01	[7.674 7.581]	[0.9990 0.5859]	2.197	-
D	10	1	0.01	[24.240 7.613]	[0.9971 0.5863]	0.695	-
E	10	0.1	0.01	[28.978 2.968]	[0.9906 0.8433]	0.220	-
F	10	0	0.01	[30.799 0.963]	$[0.9710 \pm 0.0283i]$	0.052	4.2 %
G	10	0.01	0.01	[30.476 1.327]	$[0.9614 \pm 0.0103i]$	0.077	$<$ 0.1 %
H	10	0.02	0.01	[30.225 1.609]	[0.9772 0.9304]	0.101	-

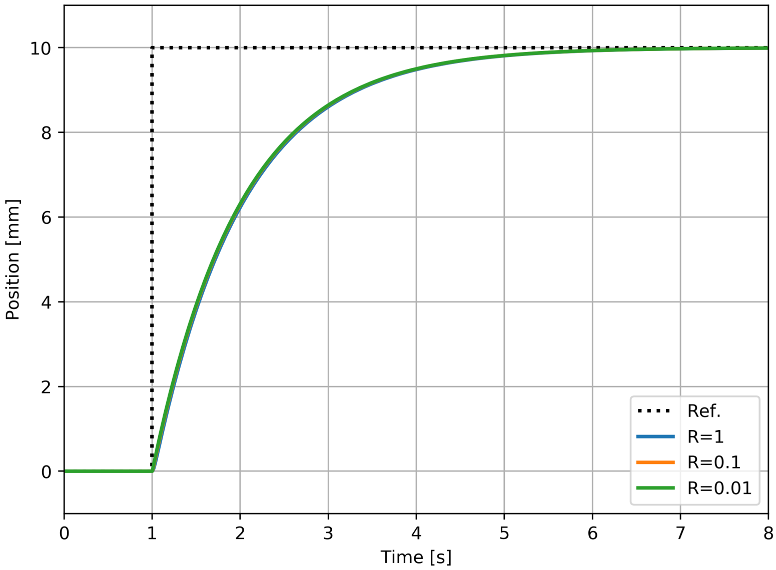


Figure 18: LQR position controller step response simulation varying the R weight.

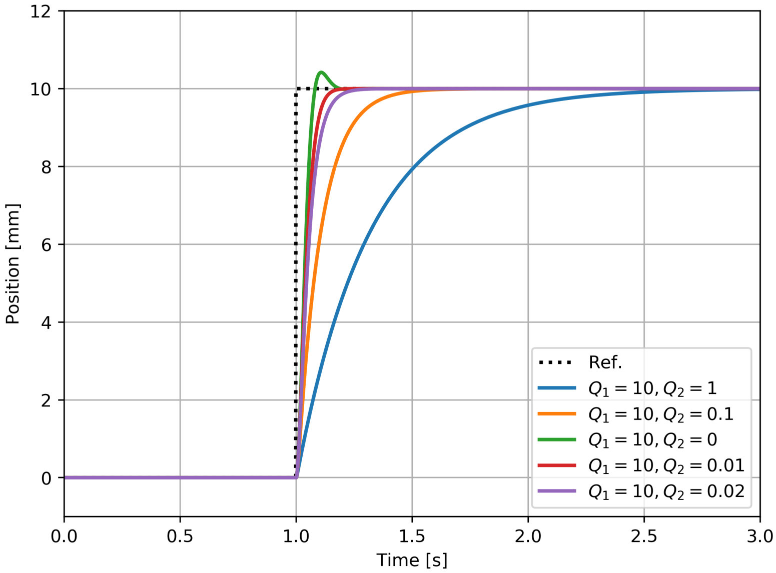


Figure 19: LQR position controller step response simulation varying the Q weight matrix, $R = 0.01$.

2.9.3 Experimental Tuning and Results

The same LQR weight configurations are then used to test the experimental setup system: the step responses are presented in Figure 20 for the choice of the input weight R and Figure 21 for the choice of q_1, q_2, weights on the state; the rise-times and overshoots are reported in Table 7.

The experimental system is tested using the same weights configurations of the simulation. For the choice of the R weight, the obtained results are very similar to the simulation ones Figure 18, supporting the model of the motor used for the controller design and simulation. Regarding the choice of q_1, q_2, compared to the simulation, the experimental runs present a more pronounced response overshoot for $q_2 = 0,\ 0.01$, Test F,G, that goes respectively from the 4.2% to the 5.8% and from <0.1% to the 0.5%, Figure 21.

For this reason the choice of the state weight matrix falls back to $q_1 = 10,\ q_2 = 0.02$: these weights guarantee that the response to step reference is still fast ($t_r = 128\,ms$) while avoiding any overshooting.

Table 7: LQR Position Controller – Experimental Bench

Test	Q_1	Q_2	R	K	T_r [s]	Overshoot
A	1	1	1	[0.976 0.888]	2.086	-
B	1	1	0.1	[2.911 2.819]	2.087	-
C	1	1	0.01	[7.674 7.581]	2.046	-
D	10	1	0.01	[24.240 7.613]	0.789	-
E	10	0.1	0.01	[28.978 2.968]	0.261	-
F	10	0	0.01	[30.799 0.963]	0.076	5.8 %
G	10	0.01	0.01	[30.476 1.327]	0.107	0.5 %
H	10	0.02	0.01	[30.225 1.609]	0.128	-

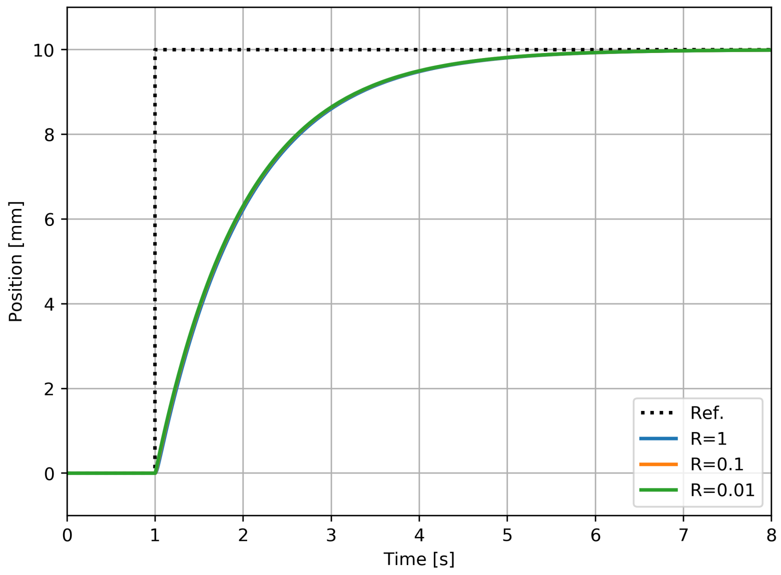


Figure 20: LQR position controller step response varying the R weight.

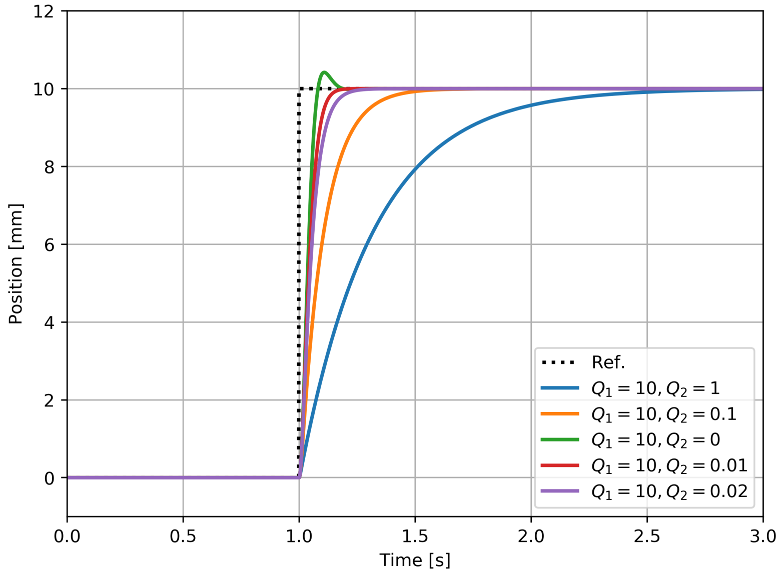


Figure 21: LQR position controller step response varying the Q weight matrix, $R = 0.01$.

2.9.4 Force Controller Design and Stability Analysis

As the name implies, the force controller objective is to mirror the (estimated) forces between the master and slave systems to obtain the overall bilateral forces balance, thus providing haptic feedback between the environment and the operator, with the already discussed user sensibility benefits. Because the master and slave environment configurations differs, a standalone force controller would yield an unstable system and so it is always coupled with a position controller to achieve a stable configuration and to obtain the position tracking.

As briefly presented in the initial section, the design of the force controller – that is the choice of K_f – is a tradeoff between the system stability and the faithful force reproduction, in particular a better operationality P_o. By increasing the force gain K_f, the force controller has more influence over the position one, monopolizing the system behavior.

Bilateral Controller Model

The bilateral controller can be described as

$$\begin{bmatrix} s^2 x_m^{ref} \\ s^2 x_s^{ref} \end{bmatrix} = Q^{-1} \begin{bmatrix} C_p(x_m - x_s) \\ C_f(\hat{F}_m^{dis} + \hat{F}_s^{dis}) \end{bmatrix} \tag{51}$$

where Q^{-1} is the inverse Quarry matrix, C_p, C_f the position and force controllers. With the disturbance compensation provided by the DOB, the mechanical motor can be modeled as the nominal one plus the disturbances outside the estimation bandwidth g_{dob}

$$x = \frac{1}{M_n s^2} \left\{ K_{t,n}\ i + \hat{F}_{dis} \frac{s}{s + g_{dob}} \right\} \tag{52}$$

The stability of the bilateral controller can be assessed by studying the stability margin of the loop transfer function between the motor and environment positions like presented in [13]. The external disturbance is modeled as

$$F_l = Z^{env}(x - x^{env}) + F^{ext} + \Delta F \tag{53}$$

where

- $Z^{env} = K + Ds$ is the environment impedance;
- x^{env} the environment edge position;
- F^{ext} the sum of external forces not dependent on the position;
- ΔF the DOB force estimation error.

and the bilateral control master F_m^{dis} and slave F_s^{dis} disturbances can be expressed as

$$F_m^{dis} = \frac{K_{tn,m}}{K_{tm}} \left\{ Z_m^{env}(x_m - x_m^{env}) + F_m^{ext} + \Delta F_m + \Delta M_m' s^2 x_m \right\} \tag{54}$$

$$F_s^{dis} = \frac{K_{tn,s}}{K_{ts}} \left\{ Z_s^{env}(x_s - x_s^{env}) + F_s^{ext} + \Delta F_s + \Delta M_s' s^2 x_s \right\} \tag{55}$$

where $\Delta M'$ denote the mass modeling error

$$\Delta M' = M - \frac{K_{tn}}{K_t} M_n \tag{56}$$

The control block diagram of the bilateral controller considering the environmental impedance and modeling errors is then presented in Figure 22 where 4 transfer functions are highlighted:

- G_{mm} is the t.f. between master disturbance and master motor position,
- G_{ss} is the t.f. between slave disturbance and slave motor position,
- G_{ms} is the t.f. between master disturbance and slave motor position,
- G_{sm} is the t.f. between slave disturbance and master motor position.

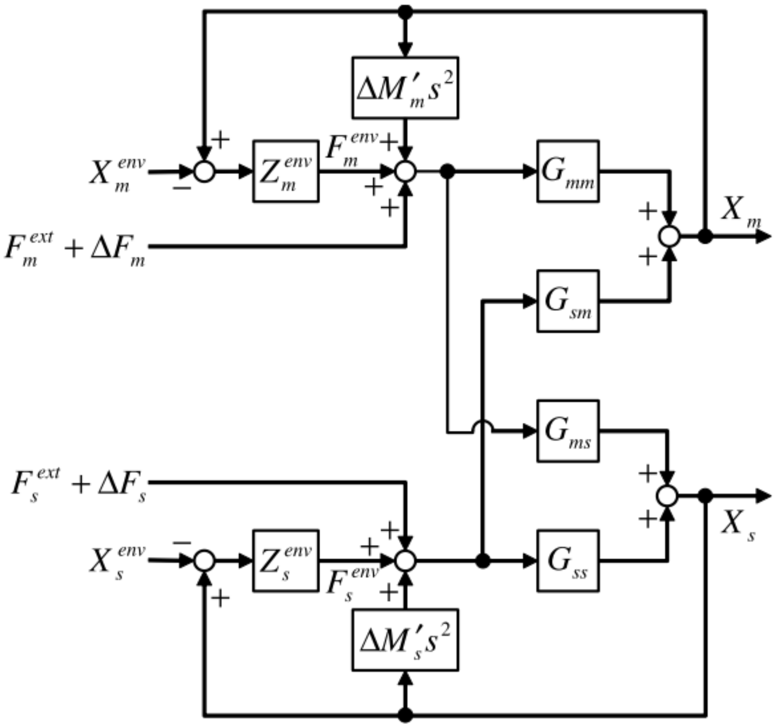


Figure 22: Bilateral controller block diagram considering environmental impedance and modelling errors [13].

Using Figure 22 [13] we can interpret the bilateral controller as having as input x, as output x^{env} and as gain the environment impedance. The stability of the system can be assessed by "opening" the loop and analyzing the loop transfer function, that can be obtained from Figure 22:

$$\begin{aligned} G_{loop} = & -G_{mm}(Z_m^{env} + \Delta M'_m s^2) \\ & -G_{ss}(Z_s^{env} + \Delta M'_s s^2) \\ & +(G_{mm}G_{ss} - G_{ms}G_{sm})(Z_m^{env} + \Delta M'_m s^2)(Z_s^{env} + \Delta M'_s s^2) \end{aligned} \tag{57}$$

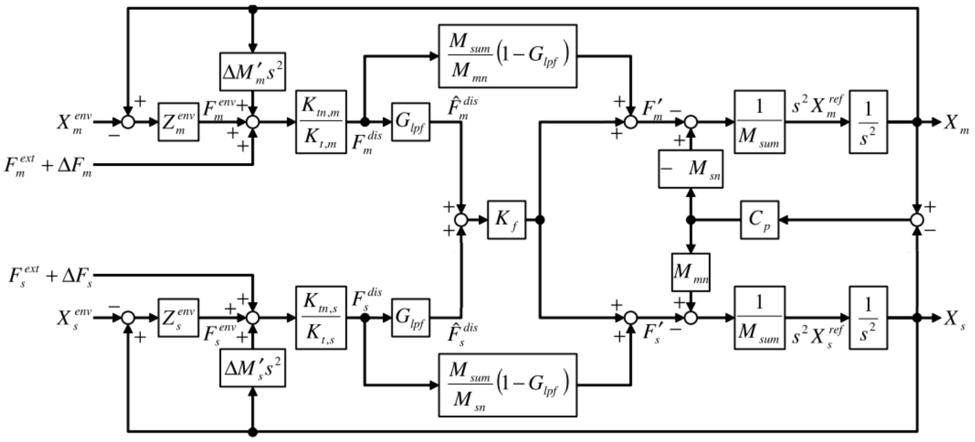


Figure 23: Equivalent bilateral controller block diagram.

The complete control loop is presented in Figure 23, with $M_{sum} = M_{mn} + M_{sn}$ and G_{lpf}

$$G_{lpf}(s) = \frac{g_s}{s + g_s}\frac{g_{dob}}{s + g_{dob}}$$

from it we can obtain:

$$G_{mm} = -\frac{K_{tn,m}}{M_{sum}s^2\, K_{t,m}}\left\{K_f G_{lpf} + (1 - G_{lpf})\left(1 + \frac{s^2}{s^2 + C_p}\right)\right\} \tag{58}$$

$$G_{ss} = -\frac{K_{tn,s}}{M_{sum}s^2\, K_{t,s}}\left\{K_f G_{lpf} + (1 - G_{lpf})\left(1 + \frac{s^2}{s^2 + C_p}\right)\right\} \tag{59}$$

$$G_{ms} = -\frac{K_{tn,m}}{M_{sum}s^2\, K_{t,m}}\left\{K_f G_{lpf} + (1 - G_{lpf})\left(1 + \frac{C_p}{s^2 + C_p}\right)\right\} \tag{60}$$

$$G_{sm} = -\frac{K_{tn,s}}{M_{sum}s^2\, K_{t,s}}\left\{K_f G_{lpf} + (1 - G_{lpf})\left(1 + \frac{C_p}{s^2 + C_p}\right)\right\} \tag{61}$$

Using the values in Table 8 for the calculation of G_{mm}, G_{ss}, G_{ms}, G_{sm} and then G_{loop} with the (57), it is possible to study the system stability by studying the open-loop transfer function G_{loop} stability margin. The stability margin is determined in MATLAB using the function `margin` for different values of K_f. The relationship between the force controller gain K_f and the stability margin is reported in Figure 24, in presence of different environment conditions between master and slave side. As previously anticipated, the choice of the force controller gain is a tradeoff between the system stability and better operatibility, defined as

$$\text{operatibility} \qquad P_o = \frac{s^2}{K_f}$$

A conservative choice is $K_f = 2\,\text{m}/\text{Ns}^2$, that offers a stability margin of 81.6 deg.

Table 8: Force control stability analysis conditions

Symbol	Description	Value
$M_{m,n}$	Master nominal mass	0.6 kg
$M_{s,n}$	Slave nominal mass	0.6 kg
M_m	Master mass	0.580 kg
M_s	Slave mass	0.575 kg
$K_{tm,n}$	Master nominal force constant	33 N/A
$K_{ts,n}$	Slave nominal force constant	33 N/A
K_{tm}	Master force constant	31.8 N/A
K_{ts}	Slave force constant	31.8 N/A
g_s	Cutoff frequency of pseudo derivative	600 rad/s
g_s	Cutoff frequency of DOB	600 rad/s
K_m	Virtual master stiffness coefficient	800 N/m
K_s	Virtual slave stiffness coefficient	2000 N/m
D_m	Virtual master damping coefficient	200 Ns/m
D_s	Virtual slave damping coefficient	100 Ns/m

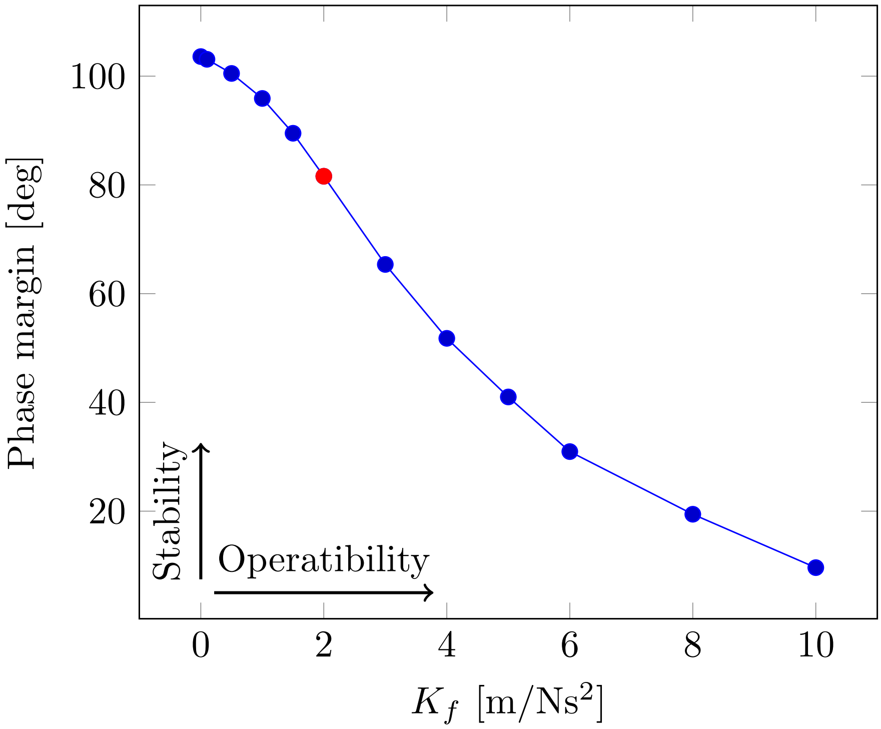


Figure 24: Bilateral control open loop transfer function phase margin for different values of K_f.

2.10 BILATERAL CONTROL TESTING

Once the bilateral controller was designed, the configuration was tested both in simulation and on the experimental setup.

2.10.1 Bilateral Control Simulation

The simulation of the bilateral system is performed using two identical motor models for the master and slave systems, both obtained from the discretization with $T = 0.1\,\mathrm{ms}$ of the continuous motor model presented in Figure 10. The bilateral controller architecture is the same as the one presented in section 2.3, with the position controller obtained from the LQR design and the force controller gain $K_f = 2$. The operator interaction is emulated by a $10\,\mathrm{N}$ constant external force on the master motor during $1\,\mathrm{s} \leqslant t \leqslant 6\,\mathrm{s}$, Figure 25. A obstacle having impedance $K = 400\,\mathrm{N/m}$ and damping $D = 50\,\mathrm{Ns/m}$ is placed $10\,\mathrm{cm}$ away from the home position. Finally the motor friction is simulated as a static $F_{st} = 0.25\,\mathrm{N}$ term and the viscous friction coefficient $B = 3.7\,\mathrm{Ns/m}$, the same values estimated for the experimental setup.

The two systems evolution is presented in Figure 25: after the external force is applied to the master, the bilateral controller accelerates the two motors to obtain the external force balance while mirroring the motor positions. The motors speed up and, as the viscous friction increases, the acceleration is slowed down. When the slave motor comes into contact with the target object a force spike (caused by the target damping) slows down the systems that reach the equilibrium at

$$\Delta x_s = \frac{F_m}{K} = \frac{10\,\mathrm{N}}{400\,\mathrm{N/m}} = 0.025\,\mathrm{m}$$

The limited Disturbance Observer bandwidth ($600\,\mathrm{rad/s}$) smooths the estimated force spike. After the external force acting on the master motor is removed, $t > 6\,\mathrm{s}$, the environment stiffness pushes the slave (and the master) back to the contact point.

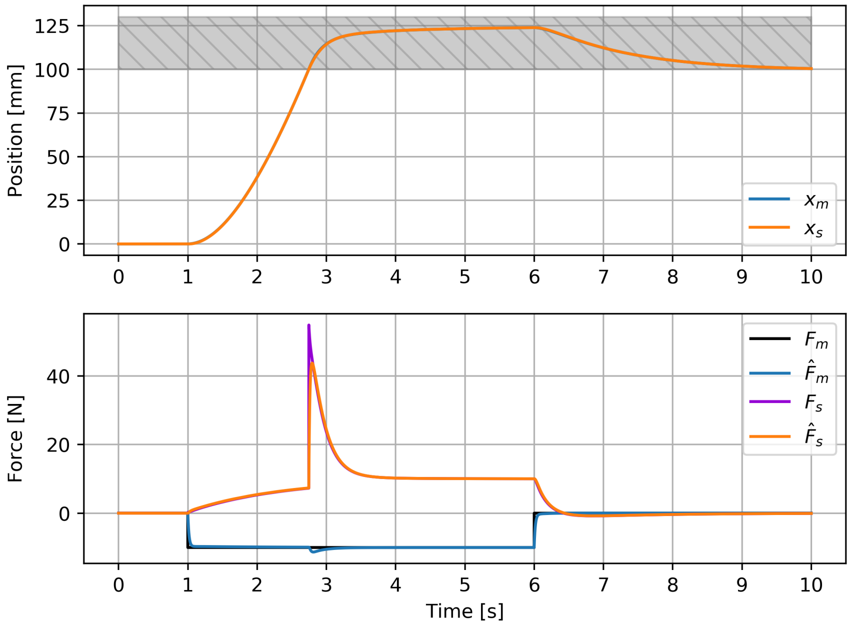


Figure 25: Bilateral control system simulation – Interaction with the environment

2.10.2 Bilateral Control Experimental Validation

For the experimental validation, the bilateral controller is implemented as a real-time C program, running on the hardware-in-the-loop computer. The operator handles the master side while the slave motor interacts with the environment, which consists of a metal object and a soft sponge. In Figure 26 the position of the two systems is presented as well as the estimated external disturbances; the position of the two obstacles are represented as a gray box (hard material) that is then swapped for a orange one (soft material) in the second part of the experiment. From the operator point of view, the interaction with the environment feels natural, as if the two systems were mechanically linked, realizing the **transparency** property between the perceived environment impedance and the actual impedance at the slave side. The force signals contain high-frequency noise that enters the control loop from the double position derivative. A simple metric to evaluate the bilateral system performance is the maximum absolute position error between the two systems

$$\max |e_p| = \max |x_m - x_p| = 0.11 \text{ mm}$$

that validate the design process choices.

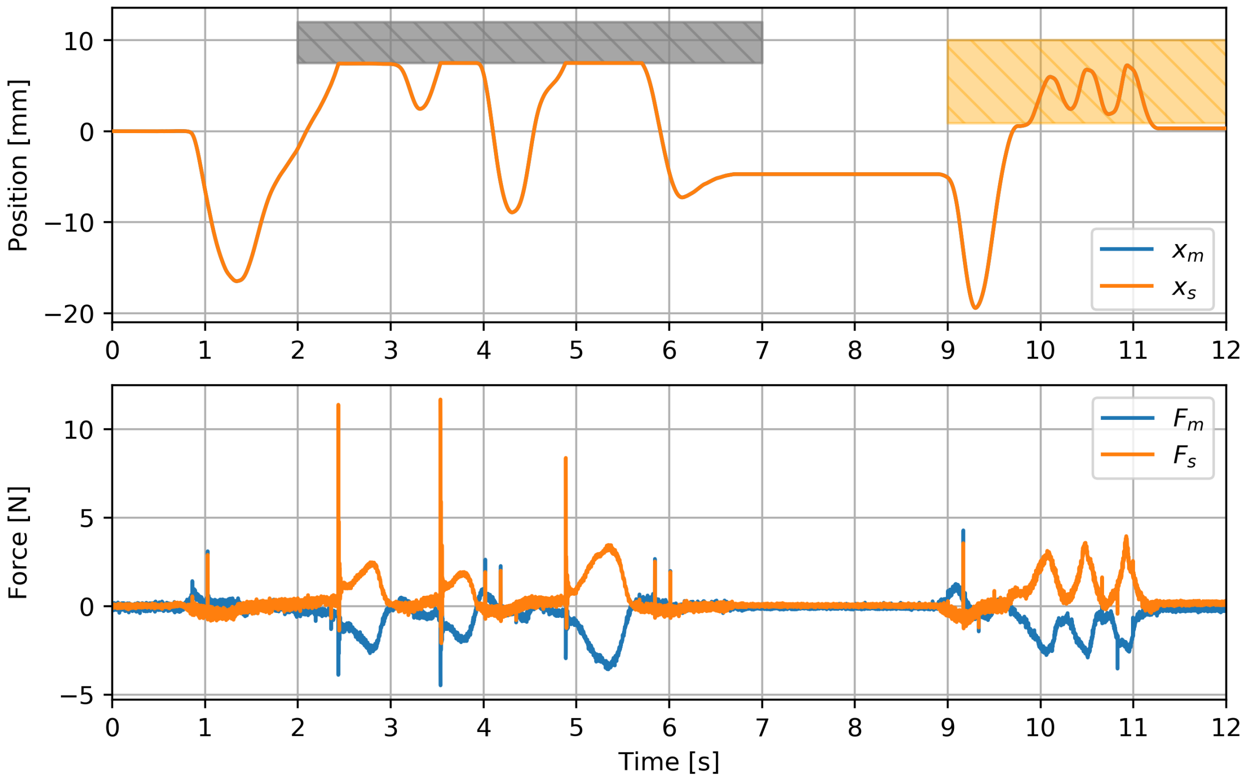


Figure 26: Bilateral control system – Interaction with the environment

2.11 MOTION COPYING SYSTEM IMPLEMENTATION

In this section it is described how the motion copying system is implemented in the experimental test bench. The MCS operation is divided into two phases, Figure 27:

1. **Motion Saving** during the operator motion the position, velocity and estimated applied force $\left[x_m, \dot{x}_m, \hat{F}_m\right]$ are saved to a *motion database*;
2. **Motion Loading** the saved task can be reproduced using the master position, velocity and force signals as a *virtual master* in the bilateral controller.

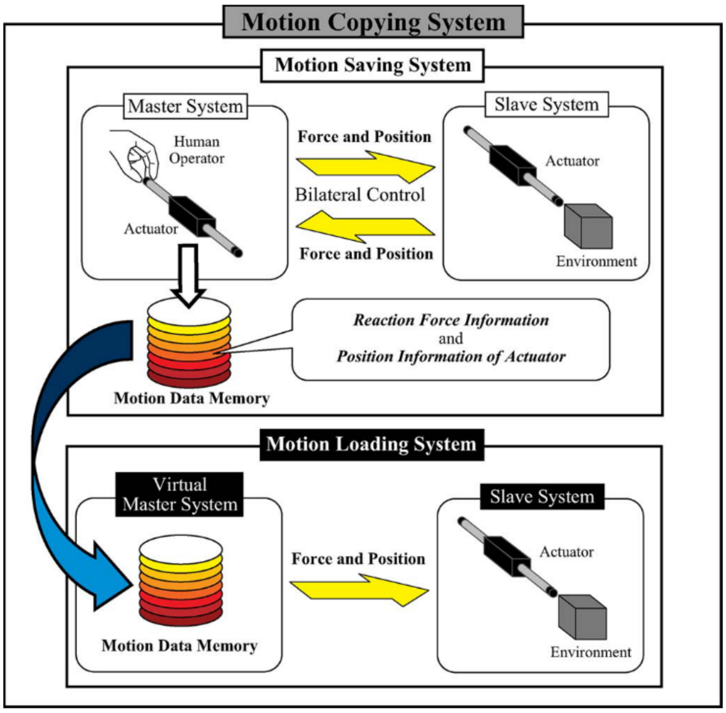


Figure 27: Bilateral Control Motion Saving Loading [5]

From the programming standpoint the motion copying system code is constituted of two parts:

- a real-time component that, during the bilateral control operation, sends the master position, speed and estimated force to a RTAI data buffer;
- a non-RT component that, once the bilateral operation is concluded, saves the sample from the data buffer to disk.

The high-frequency control loop execution (10 kHz) generates a substantial amount of data, that would quickly saturate the data buffer if saved entirely, limiting the maximum task duration. Saving the all the data sample would also be superfluous, because the operator movements don't need such high frequency sampling to be completely described. The operator hand movements properties will analyzed in Equation 4.2.

For the presented reasons, the original master references are downsampled before being sent to the data buffer, in particular they are *decimated* reducing the sampling rate to 1 kHz. Attention must be payed in the choice of the real-time application buffer STACK allocation space: the saved position, speed and force are in fact saved in the controller stack until the action recording ends, avoiding the delays introduced by the I/O operations. The `struct` used to save the master reference is composed by 3 `double` fields (x_m, $\dot{x}_m$, F_m), for a total of

$$3\ \texttt{sizeof(double)} = 24\ \texttt{sizeof(char)} = 24\ \text{byte/sample}$$

given the fact that double precision floating-point numbers have an allocation size of 4 bytes. The master reference samples are saved every 1 ms, producing a data stream of 24 kB/s. For the experimental tests the allocated stack space is 4 MB, that ensures a runtime of

$$t = \frac{4 \cdot 1024}{24} \simeq 170\,\mathrm{s}$$

To avoid the risk of a stack overflow, the data saving loop ends automatically after having recorded 120k data samples (equivalent to 2 min). The downsampling ratio 10:1 between controller loop frequency and data recording frequency can be changed to better address the trade-off between the task reproduction precision and data allocation size.

2.11.1 Signal Interpolation

During the task reproduction, we need to generate the position, speed and force signals for the bilateral controller (10 kHz) from the saved (downsampled) task samples. The first crude implementation used the same saved sample for 10 control steps: this solution introduces position, speed and force step reference discontinuities that, even if it could be acceptable for this particular configuration given the high sample rate, is not ideal. To obtain a more robust solution in the case of different downsampling ratios (e.g. 1:20, 1:50):

- a third order polynomial interpolation between position/speed samples is used to guarantee the velocity signal continuity;

- a first order (linear) interpolation is used for the force signal.

Denoting with t_k the k-th saved sample time ($f_k = 1\,\mathrm{kHz}$) and with j the control step ($f_j = 10\,\mathrm{kHz}$),

$$x_m(t_j) = a_3(t_j - t_k)^3 + a_2(t_j - t_k)^2 + a_1(t_j - t_k) + a_0 \tag{62}$$
$$F_m(t_j) = b_1(t_j - t_k) + b_0 \tag{63}$$

where the coefficients a_0, a_1, a_2, a_3 and b_0, b_1 are chosen to ensure position, velocity and force continuity:

$$\begin{aligned}
x_m(t_k) &= a_0 \\
x_m(t_{k+1}) &= \frac{a_3}{f_k^3} + \frac{a_2}{f_k^2} + \frac{a_1}{f_k} + a_0 \\
\dot{x}_m(t_k) &= a_1 \\
\dot{x}_m(t_{k+1}) &= \frac{3\,a_3}{f_k^2} + \frac{2\,a_2}{f_k} + a_1 \\
F_m(t_k) &= b_0 \\
F_m(t_{k+1}) &= b_0 + \frac{b_1}{f_k}
\end{aligned}$$

The main requirement and limiting factor of the MCS is that it is assumed that the environment remains unchanged between the saving and the loading phase. If this is not the case there will be a mismatch between the saved impedance (maser position force trajectories) and the actual environment impedance; the system behavior under a impedance mismatch is analyzed in the next section.

2.12 MCS TASK REPRODUCTION IN PRESENCE OF ENVIRONMENTAL CHANGES

In the case of a mismatch between the saved position and force relationship and the actual environment impedance the bilateral controller doesn't perfectly reproduce the saved motion. In fact, the MCS motion reproduction (virtual) master system is "read-only", so the position and force controllers compete for the achievement of conflicting goals: the position controller tries to reduce the position/speed error while the force controller drives the motors to minimize the mismatch between the saved force and the actual environment reaction force.

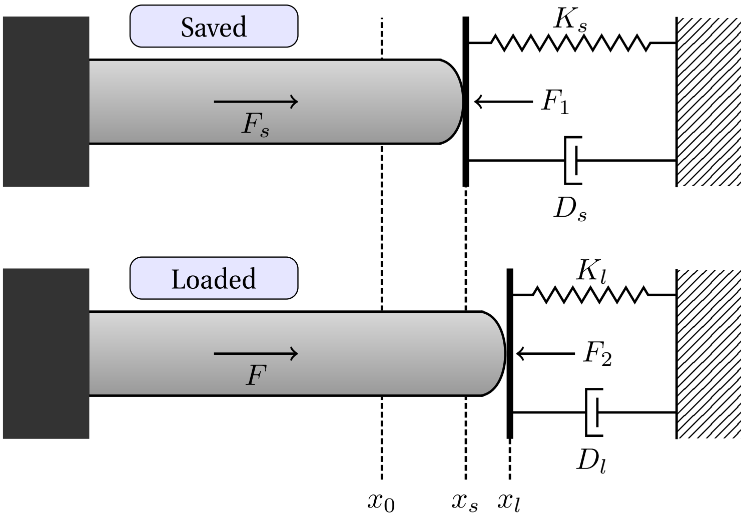


Figure 28: Motion copying system with different environment impedances.

The described situation is represented in Figure 28, where x_0 is the target edge position before the contact, x_s the saved master position, x_l the end-effector during the motion reproduction. The saved master side force reference F_s is equal to the environment reaction force during the bilateral control conditions

$$-F_s = F_1 = K_s(x_s - x_0) + D_s\dot{x}_s$$

During the reproduction phase the environment $(K_l,\ D_l)$ has changed and the reaction force is now

$$F_l = F_2 = K_l(x_l - x_0) + D_l\dot{x}_l$$

During the motion reproduction, from the force equilibrium $F_s = F_2$ between motor thrust and environmental reaction force

$$M_n\left[K_p(x_s - x_l) + K_v(\dot{x}_s - \dot{x}_l) + K_f(\hat{F}_s + \hat{F}_l)\right] = F_2 \tag{64}$$

replacing the *load* values $(K_l,\ D_l)$ and the definitions of $F_s,\ F_l$

$$\begin{aligned} x_l &= x_s + \Delta x \\ \dot{x}_l &= \dot{x}_s + \Delta\dot{x} \\ K_l &= K_s + \Delta K \\ D_l &= D_s + \Delta D \end{aligned}$$

we obtain the force equilibrium condition (66):

$$\begin{aligned} M_n[-K_p\Delta x - K_v\Delta\dot{x} + K_f(K_l\Delta x + D_l\Delta\dot{x} + \Delta K x_l \\ + \Delta D_l\dot{x}_l + \Delta K\Delta x + \Delta D_l\Delta\dot{x})] = K_l(x_l - x_0) + D_l\dot{x}_l \end{aligned} \tag{65}$$

ignoring the $\Delta K\Delta x$ and $\Delta D_l\Delta\dot{x}$ terms

$$\begin{aligned} M_n[-K_p\Delta x - K_v\Delta\dot{x} + K_f(K_l\Delta x + D_l\Delta\dot{x} + \Delta K x_l \\ + \Delta D_l\dot{x}_l)] \approx K_l(x_l - x_0) + D_l\dot{x}_l \end{aligned} \tag{66}$$

From the (66) is clear that, as it could easily be expected, the resulting motion reproduction accuracy depends on the particular design of the position and force controllers, as well as the environment changes $(\Delta K,\ \Delta D)$. During the static equilibrium $(\dot{x}_s = \dot{x}_l = 0)$ the condition simplifies to:

$$M_n[-K_p\Delta x + K_f(K_l\Delta x + \Delta K x_l + \Delta K\Delta x)] \approx K_l(x_l - x_0) \tag{67}$$

so the reproduced task precision is subjected to many factors of both the bilateral controller $(K_p,\ K_v,\ K_f)$ and the environment $(K_s - K_l,\ D_s - D_l,\ x_s - x_l)$.

2.13 MCS TASK REPRODUCTION STABILITY

In the last section we saw that the motion copying system behavior during the task reproduction depends on the environment impedance and the bilateral controller design: there are no guarantees that the motion copying

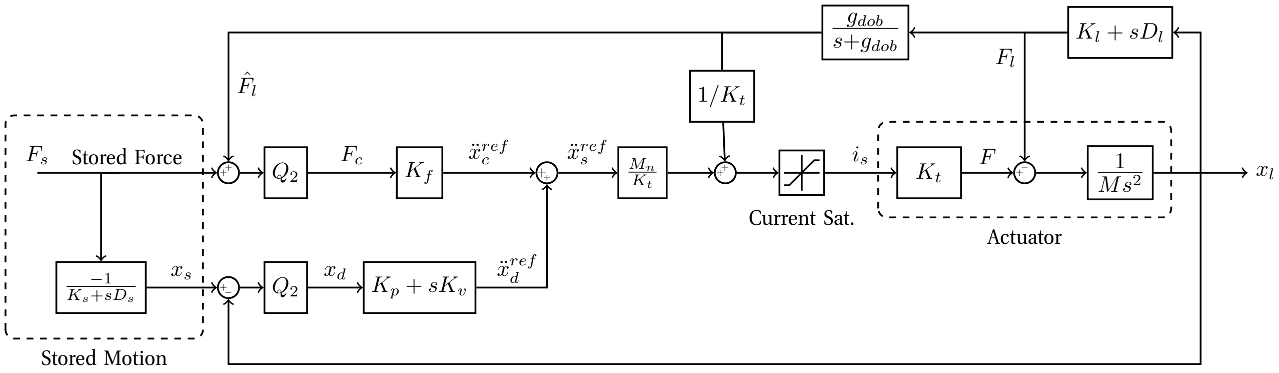


Figure 29: Motion Copying System equivalent control schematic.

system is stable under environment changes. In [5] the MCS stability is analyzed studying the pole positions of the transfer function between the saved task force and the actual one under different environment parameter variations.

The same method is applied to the studied MCS; the control block diagram is presented in Figure 29.The DOB equivalent force estimation is modeled as a simple first-order low-pass filter of bandwidth g_{dob}. With the help of MATLAB, the transfer function from the *loaded* and *saved* force can be derived:

$$G(s) = \frac{F_l}{F_s} = \frac{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0}{b_5 s^5 + b_4 s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0} \tag{68}$$

where

$$\begin{aligned}
a_0 &= -K_l \beta g_{dis}^2 \\
a_1 &= -K_l \left(2\beta g_{dis} + \alpha g_{dis}^2\right) - D_l \left(\beta g_{dis}^2\right) \\
a_2 &= -K_l \left(\beta + 2\alpha g_{dis}\right) - D_l \left(2\beta g_{dis} + \alpha g_{dis}^2\right) \\
a_3 &= -K_l \alpha - D_l \left(\beta + 2\alpha g_{dis}\right) \\
a_4 &= -D_l \alpha \\
b_0 &= K_s \eta g_{dis} \\
b_1 &= K_s \left(\delta g_{dis} + \eta + \epsilon g_{dis}\right) + D_s \eta g_{dis} \\
b_2 &= K_s \left(\delta + \gamma g_{dis} + \epsilon\right) + D_s \left(\delta g_{dis} + \eta + \epsilon g_{dis}\right) \\
b_3 &= K_s \left(\gamma + 2 g_{dis}\right) + D_s \left(\delta + \gamma g_{dis} + \epsilon\right) \\
b_4 &= 2K_s + D_s \left(\gamma + 2 g_{dis}\right) \\
b_5 &= 2D_s
\end{aligned}$$

$$\begin{aligned}
\alpha &= K_f D_s + K_v \\
\beta &= K_f K_s + K_p \\
\gamma &= \frac{2D_l}{M} + K_v + 2g_{dis} \\
\delta &= K_f g_{dis} D_l + K_p \\
\epsilon &= K_v g_{dis} + \frac{2K_l}{M} \\
\eta &= \left(K_f K_l + K_p\right) g_{dis}
\end{aligned}$$

with K_p, K_v the proportional and derivative gains of the position controller, K_f of the force controller, $K_t = 33\,\mathrm{N/A}$ the nominal force/current constant, $M_n = 0.6\,\mathrm{kg}$ the nominal motor mover mass, $g_{dob} = 600\,\mathrm{rad/s}$ the cut-off frequency of the DOB low-pass filter. The stability of the control system during the motion reproduction can be assessed studying the transfer function $G(s)$ (68) [5]. In particular the transfer function poles are calculated for different configurations of K_s, K_l, D_l. The introduced simplifications (ideal speed derivation, constant nominal parameters) are valid for movements and motor dynamics completely captured by the DOB bandwidth, where the DOB disturbance rejection is maximum and the controller system acts as the nominal one.

2.13.1 Soft Target Stability

In this simulation it is hypothesized that the environment conditions during the bilateral task recording were $K_s = 300\,\mathrm{N/m}$, $D_s = 50\,\mathrm{Ns/m}$, while during the reproduction the target stiffness is varied between $K_l \in [0,\ 10^5]\ \mathrm{N/m}$ and the damping between $D_l \in [0,\ 10^3]\ \mathrm{Ns/m}$. The position evolution of the system poles is reported respectively in Figure 30 and Figure 31.

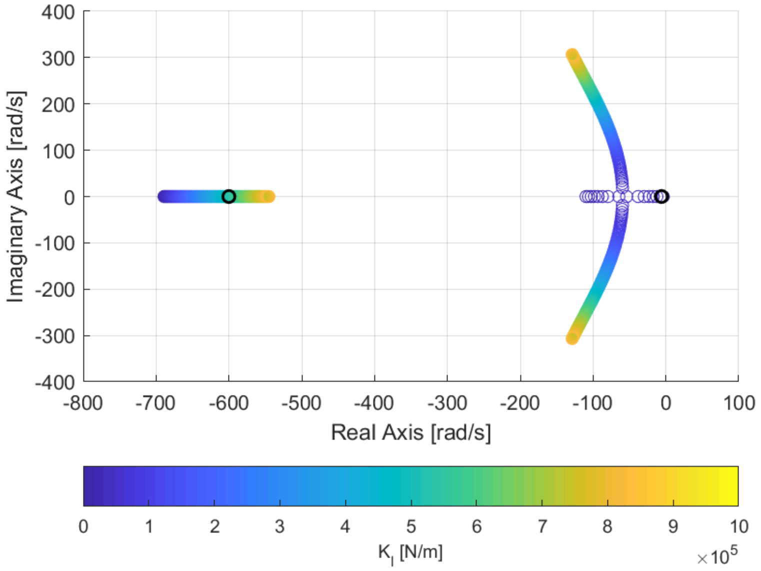


Figure 30: System poles with $K_s = 300\,\mathrm{N/m},\ D_s = 50\,\mathrm{Ns/m}$ – $K_l \in [0,\ 10^5]\ \mathrm{N/m}$

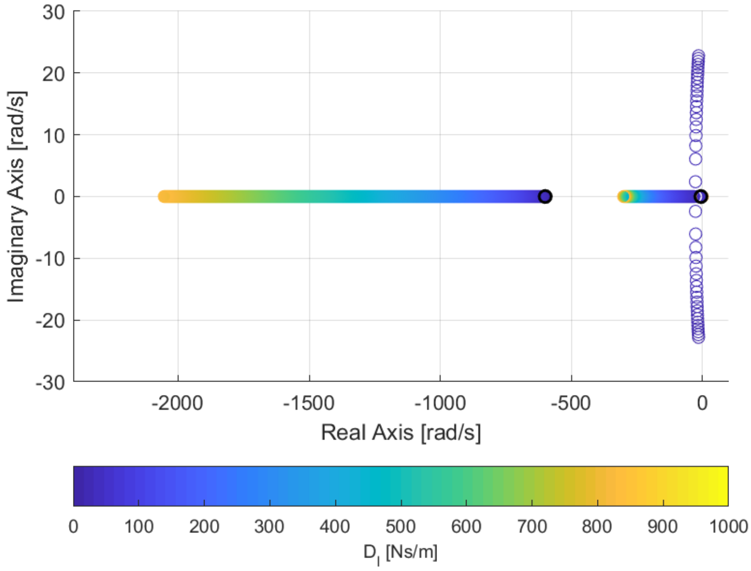


Figure 31: System poles with $K_s = 300\,\mathrm{N/m},\ D_s = 50\,\mathrm{Ns/m}$ – $D_l \in [0,\ 10^3]\ \mathrm{Ns/m}$

STIFFNESS VARIATIONS The position of the 5 system poles is analyzed as a function of the environment stiffness, that is varied between $K_l = 0 \rightarrow 10^5$ N/m:

- two poles in $p_1 = -600\,rad/s$ and $p_2 = -6\,rad/s$ are invariant to K_l changes and are highlighted with a dark "∘" in the root locus;
- a pole transition from $p_3 = -689\,rad/s$ to $-545\,rad/s$;
- two poles, starting from $p_4 = -109\,rad/s$ and $p_5 = -3.6\,rad/s$ become complex-conjugate for $K_l \simeq 1990\,N/m$, then approaching $p_{3,4} = -128 \pm 306i\ rad/s$.

DAMPING VARIATIONS The position of the 5 system poles is analyzed as a function of the environment damping, that is varied between $D_l = 0 \rightarrow 10^3$ Ns/m:

- two poles in $p_1 = -600\,rad/s$ and $p_2 = -6\,rad/s$ are invariant to K_f changes and are highlighted with a dark "∘" in the root locus,
- two complex-conjugates poles is $p_{3,4} = -15.2 \pm 22.8i\ rad/s$ transition to $p_3 = -300.7\,rad/s$ and $p_4 = -0.7\,rad/s$, becoming purely real for $D_l = 33\,Ns/m$,
- the last pole transition from $p_5 = -599\,rad/s$ to $-2052\,rad/s$.

2.13.2 Stiff Target Stability

In this simulation it is hypothesized that the environment conditions during the bilateral task recording were $K_s = 10^4$ N/m, $D_s = 50\,Ns/m$, while during the reproduction the target stiffness is varied between $K_l \in [0,\ 10^5]\ N/m$ and the damping between $D_l \in [0,\ 10^3]\ Ns/m$. The position evolution of the system poles is reported respectively in Figure 32 and Figure 33.

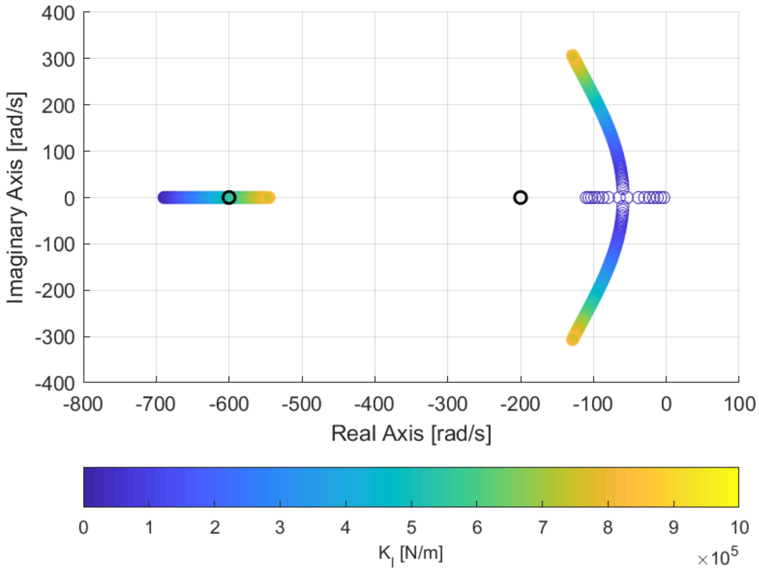


Figure 32: System poles with $K_s = 10^4$ N/m, $D_s = 50\,\text{Ns/m} - K_l \in [0,\ 10^5]$ N/m

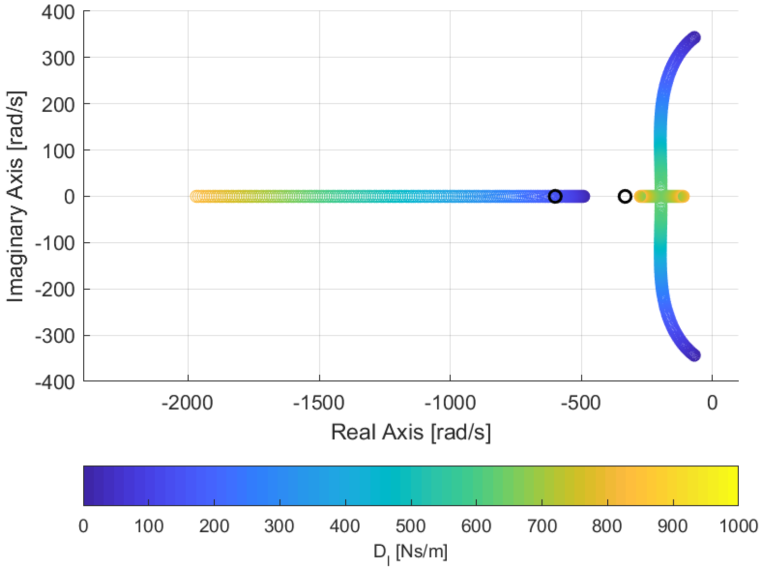


Figure 33: System poles with $K_s = 10^4$ N/m, $D_s = 50\,\text{Ns/m} - D_l \in [0,\ 10^3]$ Ns/m

STIFFNESS VARIATIONS By varying the environment stiffness the position of the 5 system poles is analyzed; as K_l goes from 0 to 10^5 $\mathrm{N/m}$:

- two poles in $p_1 = -600\,\mathrm{rad/s}$ and $p_2 = -200\,\mathrm{rad/s}$ are invariant to K_l changes and are highlighted with a dark "$\circ$" in the root locus;
- a pole transition from $p_3 = -689\,\mathrm{rad/s}$ to $-545\,\mathrm{rad/s}$;
- two poles, starting from $p_4 = -109\,\mathrm{rad/s}$ and $p_5 = -3.6\,\mathrm{rad/s}$, become complex-conjugate for $K_f \approx 1990\,\mathrm{N/m}$, approaching $-128 \pm 306i$ $\mathrm{rad/s}$ for $K_l \to 10^6$ $\mathrm{N/m}$.

DAMPING VARIATIONS By varying the environment damping the position of the 5 system poles is analyzed; as D_l goes from 0 to 10^3 $\mathrm{Ns/m}$:

- two poles in $p_1 = -600\,\mathrm{rad/s}$ and $p_2 = -333.3\,\mathrm{rad/s}$ are invariant to D_l changes and are highlighted with a dark "$\circ$" in the root locus,
- two complex-conjugate poles in $p_{3,4} = 69.4 \pm 343i$ $\mathrm{rad/s}$ transition to $p_3 = -111\,\mathrm{rad/s}$ and $p_4 = -275\,\mathrm{rad/s}$ as D_l increases, becoming purely real with $D_l = 597\,\mathrm{Ns/m}$
- the last pole p_5 transition from $-599\,\mathrm{rad/s}$ to $-2052\,\mathrm{rad/s}$.

2.13.3 Final Comments

From the system pole loci in response to the environmental condition changes, the analyzed motion copying system presents always a stable behavior: the real part of every transfer function pole is strictly negative, yielding a BIBO stable system. This result guarantees that, in the case of a mismatch between the original environmental conditions (during the operator recording) and the conditions during the task reproduction, the overall system is stable, with the downside of an imprecise task reproduction.

2.14 MOTION COPYING SYSTEM TESTING

In this chapter the motion copying system was introduced and its behavior in the presence of environmental changes was analyzed. In the following section we proceed to test the system, both in simulation and on the experimental bench.

2.14.1 Simulation Testing

The same environment used for the bilateral control is now used for the motion copying system simulation. The master position, speed and estimated force from the bilateral control simulation (Figure 25) are used as virtual master reference; the results are presented in Figure 34. The control system achieve position tracking and force mirroring (the maximum error between the virtual master reference and the end-effector motion is $0.54\,\mathrm{mm}$), successfully reproducing the bilateral control task.

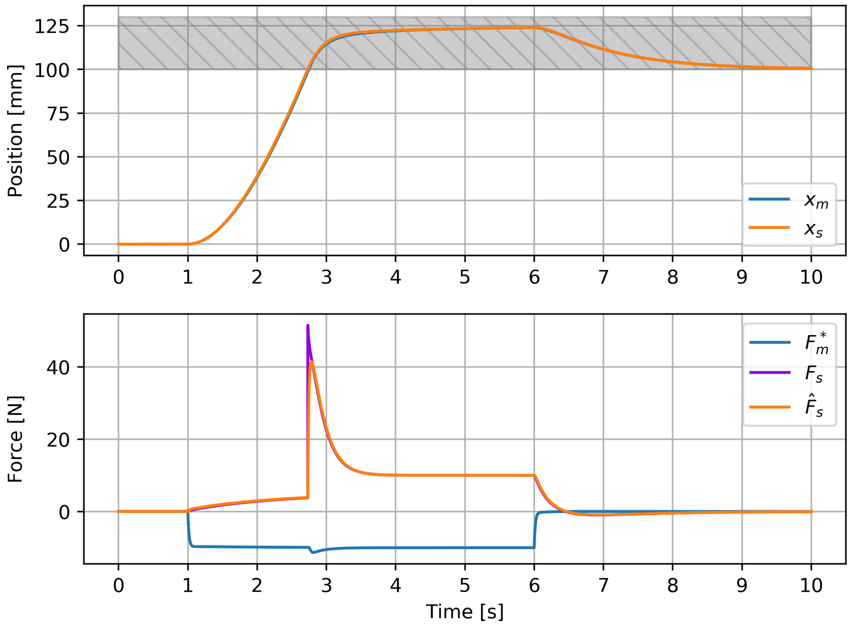


Figure 34: Motion copying system simulation testing

2.14.2 Experimental Testing

The experimental validation of the motion copying system is performed by saving an operator action, the interaction of the end-effector with a sponge at a distance of about $3\,\mathrm{cm}$ from the home position, and then reproducing said action. In Figure 35 the results are presented.

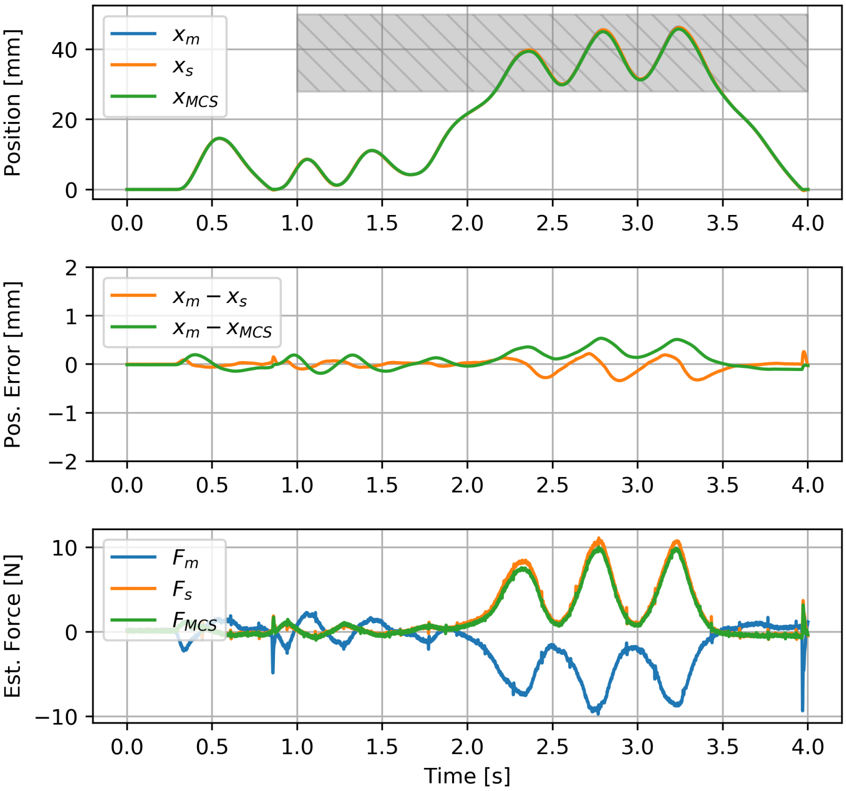


Figure 35: Motion copying system experimental testing – Task reproduction with original environment.

2.14.3 Testing with Environmental Changes

The same MCS task is reproduced when the original target was replace with a stiffer and a softer one, Figure 36 and Figure 37. As predicted the overall system behavior is stable, but because the environment impedance doesn't match the saved position/force reference are present positioning and applied force errors. In particular with the softer target there is a position overshoot and a force under-activation, instead with the stiffer target there is a position undershoot and a force over-activation.

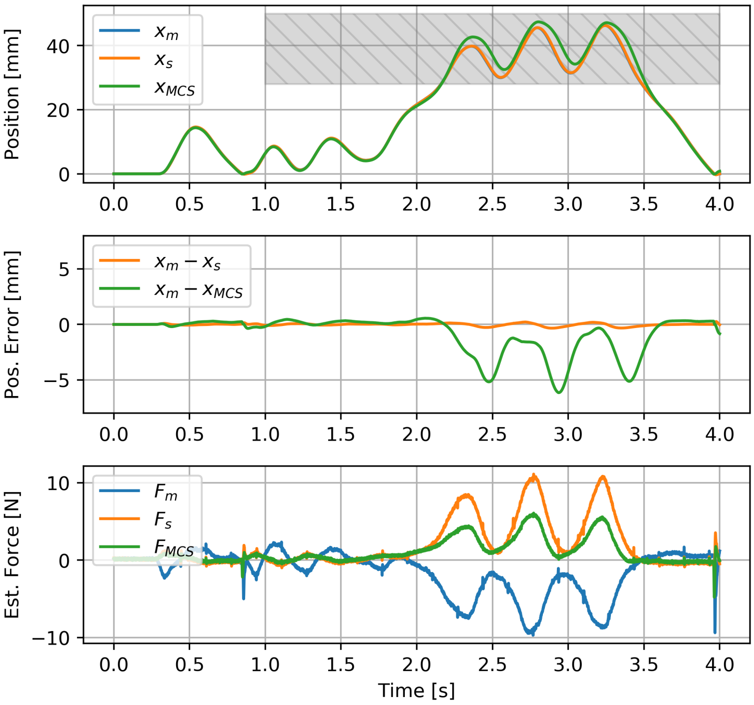


Figure 36: Motion copying system experimental testing – Task reproduction with softer environment.

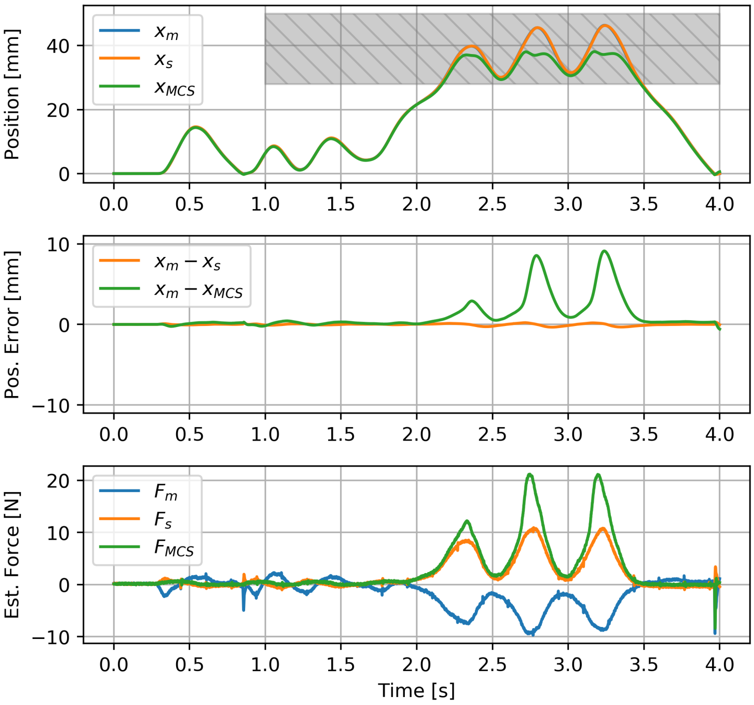


Figure 37: Motion copying system experimental testing – Task reproduction with stiffer environment.

3 DISTURBANCE AND SPEED ESTIMATION

3.1 INTRODUCTION

The bilateral controller architecture makes use of a Disturbance Observer (DOB) to estimate the total external disturbances acting on each motor, that, besides the external forces resultant, includes all the non-idealities such as friction and non-nominal mass and force constant. The DOB estimation bandwidth is selected by trial-and-error: a trade off is chosen between large bandwidths, that capture faster dynamics but are more corrupted by the quantization errors, or lower bandwidths, that present smoother estimates but attenuate the high-frequency components. The objective of this chapter is to find an optimal configuration of the DOB that offers the best tradeoff between estimation bandwidth and noise rejection.

The tool used to tune the observer is the Kalman filter.

3.2 DOB EQUIVALENT STATE-SPACE MODEL

The first step is to obtain a state-space representation of the DOB [14]: the external disturbance is modeled as an additional state variable, F_{dist}. The state vector of the system is then $\mathbf{x} = [x,\ \dot{x},\ F_{dist}]^T$, i.e. the motor position, speed and the total external disturbance force, while the the only input is the motor current reference $u = i^{ref}$ and the only measurable output is the quantized motor position x

$$\dot{\mathbf{x}}(t) = F\,\mathbf{x}(t) + G\,u(t) \tag{69}$$

$$y(t) = H\,x(t) \tag{70}$$

where

$$F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1/M \\ 0 & 0 & 0 \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ K_t/M \\ 0 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

and the system observability can be checked by verifying if the observability matrix has full rank, which it does:

$$\text{rank}\ \mathcal{O} = \text{rank} \begin{bmatrix} H \\ HF \\ HF^2 \end{bmatrix} = \text{rank} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/M \end{bmatrix} = 3 \tag{71}$$

The state observer structure is the one of a predictor-corrector:

$$\hat{\dot{\mathbf{x}}}(t) = F\,\hat{\mathbf{x}}(t) + G\,u(t) + L\{y(t) - H\mathbf{x}(t)\} \quad \text{with} \quad L = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} \tag{72}$$

The force disturbance estimation dynamics can be made explicit in the transfer function form:

$$\begin{aligned}\hat{F}_{dist}(s) &= \begin{bmatrix}0 & 0 & 1\end{bmatrix}\left\{(sI - F + LH)^{-1}\left[GU(s) + LY(s)\right]\right\} \\ &= \left[K_t U(s) - s^2 MY(s)\right]\frac{-l_3}{Ms^3 + l_1 Ms^2 + l_2 Ms - l_3} \qquad (73)\end{aligned}$$

that is the filtered disturbance obtained by subtracting the motor inertia force from the (nominal) motor force $K_t u = K_t i^{ref}$: modeling the disturbance in the state space as an unobservable input yields a structure equivalent to the DOB's, where the choice of the observer gain L is equivalent to the choice of the DOB filtered derivative bandwidths g_s, g_{dob}, but now several tools can be used to guide this choice.

3.3 OPTIMAL ESTIMATION

with output noise we mean the errors that prevent from measuring the true state, for example the encoder quantization

with process noise the equivalent noise that acts on the continuous system state, including the modeling errors

Consider a state-space process model perturbed by process and output noises: with the hypothesis that these noises probability distributions are known, the Kalman filter is an optimal state observer in the sense that it minimizes the mean squared error on the state estimation or, from the probability point of view, the filter output maximizes the $\hat{\mathbf{x}}$ joint probability distribution. Three random noises are introduced, 2 model uncertainty noises w_1, w_2 and the measurement noise v:

- w_1 acts in parallel to the system input and includes all the noises interfering with the motor control and driving, e.g. the limited DAC resolution quantization noise;
- the motor external disturbance is modeled as a random walk fed by the random distribution w_2
- the motor position is corrupted by the sampling quantization error v

All three noises are hypothesized to

- have a probability distribution with zero mean $E[n] = 0$ and variance σ^2;
- be stationary (their probability distribution doesn't change over time);
- be ergodic (the statistical average converges to the time average);
- have null autocorrelation and correlation

$$R_w(\tau) = E[w(t)w^T(t+\tau)] = 0 \;\forall\tau \neq 0$$

$$R_v(\tau) = E[v(t)v^T(t+\tau)] = 0 \;\forall\tau \neq 0$$

$$E[w(t)v^T(t)] = 0 \;\forall\tau \neq 0$$

- the covariance matrices of the noises are known

$$E[w_1(k)w_1^T(k)] = Q_{1,k} \quad E[w_2(k)w_2^T(k)] = Q_{2,k} \quad E[v(k)v^T(k)] = R_k$$

The state-space motor model that includes the introduced stochastic noises is:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= F\mathbf{x}(t) + G\left[u(t) + w_1(t)\right] + \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T w_2(t)\\ y(t) &= Hy(t) + v(t)\end{aligned}$$

where w_1 acts in parallel to the input u and w_2 directly on the force disturbance random walk. A block representation is shown in Figure 38.

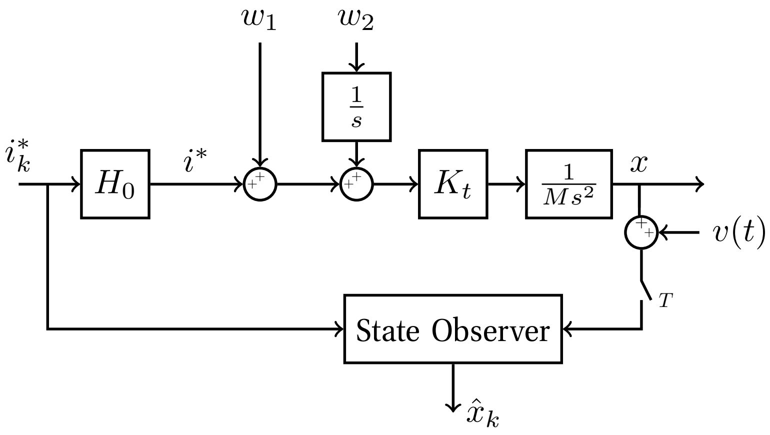


Figure 38: Motor model with the output and process noises and the State Observer

Temporary ignoring the introduced noises and following the same discretization used in the LQR controller design, the continuous time matrices (F, G) are converted to the discrete time ones (Φ, Γ). The observer has a predictor-corrector structure:

$$\begin{aligned}\hat{x}[k|k-1] &= \Phi\hat{x}[k-1|k-1] + \Gamma u(k-1)\\ \hat{x}[k|k] &= \hat{x}[k|k-1] + L[k](y[k] - H\hat{x}[k|k-1])\end{aligned}$$

in the first equation (temporal update) the state of the k-nth step is estimated using the previous step $k-1$ state estimate and the current system input, in the second one (measurement update) the estimation is corrected by weighting the **innovation** $\nu[k] = y[k] - H\hat{x}[k|k-1]$ with the $L[k]$ matrix.

3.3.1 Weighted Least Squares

Consider a linear application where the output y is corrupted by the zero-mean noise v

$$y = Hx + v \tag{74}$$

where $y \in \mathbb{R}^p$ is the output measurements vector, $x \in \mathbb{R}^n$ the unknown vector, $v \in \mathbb{R}^p$ noise vector. With $p \geqslant n$ the system is overdetermined: the number of measurement samples is greater than the state dimension. The objective of the algorithm is to find the x that minimizes the quadratic cost J, weighted by the diagonal weight matrix $W \in \mathbb{R}^{p \times p}$:

$$J = \frac{1}{2}v^T W v$$

The convex optimization problem is straightforward:

$$\begin{aligned}\frac{\partial J}{\partial x} &= (y - Hx)^T W(-H) = 0 \\ H^T W H x &= H^T W y \\ \hat{x} &= (H^T W H)^{-1} H^T W y \end{aligned} \tag{75}$$

If the noise v covariance matrix is known

$$E[vv^T] = R = \sigma^2 I$$

a direct choice for W is R^{-1}: this way the low noise output errors are weighted more than the ones coming from the more noisy outputs. The covariance error matrix of $\hat{x}$ is P (using the (74)):

$$\begin{aligned} P &= E[(\hat{x} - x)(\hat{x} - x)^T] \\ &= E[(H^T W H)^{-1} H^T W v v^T W H (H^T W H)^{-1}] \\ &= (H^T R^{-1} H)^{-1} \end{aligned}$$

that appears in the estimation (75).

3.3.2 Recursive Algorithm

As presented, the weighted least squares algorithm increases the vector dimension p at each step k. The computational load can be lightened by implementing the algorithm recursively:

$$\begin{bmatrix} y_o \\ y_n \end{bmatrix} = \begin{bmatrix} H_o \\ H_n \end{bmatrix} x + \begin{bmatrix} v_o \\ v_n \end{bmatrix}$$

where the first line include the *old* samples, the second the *new* ones. The weighted estimation becomes:

$$\begin{bmatrix} H_o \\ H_n \end{bmatrix}^T \begin{bmatrix} R_o^{-1} & \\ & R_n^{-1} \end{bmatrix} \begin{bmatrix} H_o \\ H_n \end{bmatrix} \hat{x} = \begin{bmatrix} H_o \\ H_n \end{bmatrix}^T \begin{bmatrix} R_o^{-1} & \\ & R_n^{-1} \end{bmatrix} \begin{bmatrix} y_o \\ y_n \end{bmatrix}$$

the *new* state estimation $\hat{x} = \hat{x}_o + \Delta\hat{x}$ is obtained by the *old* one and the incremental contribution by the *new* samples $\Delta\hat{x}$. Using the fact that (75)

$$(H_o^T R_o^{-1} H_o)\hat{x}_o = H_o^T R_o^{-1} y_o$$

we obtain

$$\hat{x} = \hat{x}_o + P_n H_n^T R_n^{-1} (y_n - H_n \hat{x}_o) \tag{76}$$

$$P_n = [P_o^{-1} + H_n^T R_n^{-1} H_n]^{-1} \tag{77}$$

where P_o is the state covariance matrix at the time step $k-1$ and P_n the updated matrix at time k.

3.3.3 Kalman Filter Derivation

Like in the recursive Least Squares algorithm, where $P[k]H^T R_n^{-1}$ weights the corrective action of the innovation on the state estimate, in the predictor–corrector observer the $L[k]$ matrix have a similar role in correcting the state estimate using new information that is the innovation (difference between actual and predicted system output). The main difference between the two algorithms is that the observer state is time-changing, while the Least Squares the unknown vector is not. To integrate the state evolution into the algorithm, it is introduced the state estimation covariance matrix $M[k]$ *before* the measurement update:

$$\begin{aligned} M[k+1] &= E\left[(\hat{\mathbf{x}}[k+1|k]-\mathbf{x}[k+1])(\hat{\mathbf{x}}[k+1|k]-\mathbf{x}[k+1])^T\right] \\ &= E\left[\Phi(\hat{\mathbf{x}}[k|k]-\mathbf{x}[k])(\hat{\mathbf{x}}[k|k]-\mathbf{x}[k])^T\Phi^T + \Gamma_1 w_1[k]w_1^T[k]\Gamma_1^T + \Gamma_2 w_2[k]w_2^T[k]\Gamma_2^T\right] \\ &= \Phi P[k]\Phi^T + Q_w \end{aligned} \tag{78}$$

where $Q_w = Q_{w1} + Q_{w2} = \Gamma_1 R_{w1}\Gamma_1^T + \Gamma_2 R_{w2}\Gamma_2^T$ is the covariance model error matrix. To obtain $P[k]$ we can apply the matrix inversion lemma to the (77):

$$\begin{aligned} P[k] &= [M^{-1}[k] + H^T R_v^{-1} H]^{-1} \\ &= M[k] - M[k]H^T(R_v + HM[k]H^T)^{-1}HM[k] \end{aligned} \tag{79}$$

The predictor-corrector Kalman filter is then:

1. during the time step k the next step state $\hat{x}(k+1)$ and its covariance matrix $M(k+1)$ are estimated,

$$\begin{cases} \hat{x}[k+1|k] & = \Phi\hat{x}[k|k] + \Gamma u[k] \\ M[k+1] & = \Phi P[k]\Phi^T + Q_w \end{cases}$$

2. when the $y(k+1)$ sample is available, the covariance matrix $P(k+1)$ is calculated from $M(k+1)$ and the incremental correction to the state estimate is performed using the innovation and $L[k]$

$$\begin{cases} P[k+1] & = M[k+1] - M[k+1]H^T(R_v + HM[k+1]H^T)^{-1}HM[k+1] \\ L[k+1] & = P[k+1]H^T R_v^{-1} \\ \hat{x}[k+1|k+1] & = \hat{x}[k+1|k] + L[k+1]\,(y[k+1] - H\hat{x}[k+1|k]) \end{cases}$$

Dualism Between Optimal Control and Estimation

For the hypothesis of the stationary noise probability distribution the matrices R_v, Q_w are time-invariant: it is then possible to study their steady-state evolution separately from the observer:

$$\begin{cases} P_\infty & = M_\infty - M_\infty H^T(HM_\infty H^T + R_v)^{-1}HM_\infty \\ M_\infty & = \Phi P_\infty \Phi^T + Q_w \end{cases}$$

where we can recognize a problem dual to the optimal control one:

Optimal Control	Optimal Estimation
Φ	Φ^T
Γ	H^T
M	P
Q	Q_w
R	R_v

We can obtain a time-invariant feedback matrix L by ignoring the (P, M) evolution and consider its regime value by solving the following equation:

$$P_\infty = \Phi P_\infty \Phi^T + Q_w - \Phi P_\infty H^T (H P_\infty H^T + R_v)^{-1} H P_\infty \Phi^T$$

where we can identify an ARE (Algebraic Riccati Equation) in the variable P_∞. Like it was done for the optimal controller, the ARE is solved and the innovation gain L calculated directly in MATLAB with the `dlqr` command

$$L = \mathtt{dlqr}\left(\Phi^T,\ H^T,\ Q_w,\ R_v\right)^T \tag{80}$$

3.4 FILTER TUNING

The tuning of the Kalman filter consists in finding R_v, Q_w, that express respectively the uncertainty on the system output measurements and the model noise to obtain the observer gain L with the (80).

3.4.1 Measurement Noise Covariance R_v

The evaluation of the measurement covariance R_v is done assuming that the measurement noise is caused entirely by the position encoder quantization, a random uniform distribution between $[-q/2;\ q/2]$, $q = 10^{-6}$ m:

$$R_v = \sigma^2 = \frac{q^2}{12} = 8.33 \cdot 10^{-14}\ m^2$$

3.4.2 Process Noise Covariance Q_w

Contrary to the measurement one, the process noise effects on the system cannot be directly determined as it acts on the continuous portion of the system, varying throughout one sample period. The generic process noise $w(t)$ effects on the continuous model

$$\dot{x}(t) = Fx(t) + Gw(t)$$

can be integrated to obtain the following discrete model

$$x(k+1) = \Phi x[k] + \int_0^T e^{F\eta} G w(\eta)\ d\eta$$

Using the definition of M (78) we obtain the process noise covariance matrix Q_w:

$$Q_w = \int_0^T \int_0^T e^{F\sigma} G E\left[w(\sigma) w^T(\eta)\right] G^T e^{F^T \eta} d\sigma d\eta \tag{81}$$

from the hypothesis that the noise is *white*, the correlation is not null only when $\eta = \sigma$

$$E\left[w(\sigma)w^T(\eta)\right] = R_w = \sigma_w^2\delta(\eta - \sigma)$$

where R_w is the power spectral density of the continuous white noise. Using the sampling property of the Dirac delta:

$$Q_w = \sigma_w^2 \int_0^T \int_0^T e^{F\sigma} G G^T e^{F^T\eta} d\sigma d\eta = \sigma_w^2 \mathcal{W}_{0\to T} \tag{82}$$

where $\mathcal{W}_{0\to T}$ is the *controllability Gramian*. The Q_w term consist then of the w noise variance σ_w^2 multiplied by the Gramian.

The presented motor system model includes two process noises: w_1 acting in parallel to the input and w_2 feeding the external disturbance random walk. Both their Gramians are calculated hereafter, where $\alpha = -B/M_n$, $\beta = K_{tn}/M_n$:

$$\begin{aligned}\mathcal{W}_{1,0\to T} &= \int_0^T \int_0^T e^{F\sigma} G G^T e^{F^T\eta} d\sigma d\eta \\ &= \begin{bmatrix} T^3/3\beta^2 & T^2/2\beta^2 + T^3/3\alpha\beta^2 & 0 \\ T^2/2\beta^2 + T^3/3\alpha\beta^2 & \beta^2 T + \alpha^2\beta^2 T^3/3 + 2\alpha\beta^2 T^2/2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{83}$$

$$\begin{aligned}\mathcal{W}_{2,0\to T} &= \int_0^T \int_0^T e^{F\sigma} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} e^{F^T\eta} d\sigma d\eta \\ &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & \beta^2 T^3/3 & \beta T^2/2 \\ 0 & \beta T^2/2 & T \end{bmatrix} \end{aligned} \tag{84}$$

In the previous section the duality between the optimal LQR control theory and the Kalman filter derivation was highlighted. Like the LQ controller performance depends exclusively on the ratio between the state and input weights and not their absolute value, the Kalman filter performance depends on the ratio between R_v and Q_w. The L matrix adjust the state estimation after the measure, based on how accurate the process model (Q_w) is in respect to the output noise (R_v). To tune the filter we can set R_v to the calculated (fixed) value and adjust Q_w until a satisfactory tuning is reached, exactly like it was done for the LQR controller. Without loss of generality we can impose $\sigma_w^2 = 1$ and introduce a multiplicative factor ρ to scale the individual process noises covariances. The complete process noise covariance matrix is then

$$Q_w = Q_{w1} + Q_{w2} = \rho_1 \mathcal{W}_{1,0\to T} + \rho_2 \mathcal{W}_{2,0\to T}$$

where ρ_1, ρ_2 are two "knobs" used to properly tune the filter.

3.4.3 Bartlett Whiteness Test

The Bartlett whiteness test in based on the idea of adjusting Q_w until the observer innovation $\nu = y - H\hat{x}$ has a uniform spectrum (a property of the white noise), meaning that the Kalman filter is able to extract all information from the measurements and the model. The innovation *whiteness* is measured by integrating over the frequencies the absolute value of the innovation spectrum: the obtained diagram, normalized between 0 and 1, is called **cumulative periodogram**. If the innovation is a white noise signal, it has constant spectrum and therefore its integral should be a straight line between $(0,0)$ and $(\Omega/2,1)$; if the experimental cumulative periodogram:

- is above the ideal one, the Kalman filter has a high-pass behavior and the Q_w needs to be lowered to weight more the state estimation;
- is below the ideal one, the Kalman filter has a low-pass behavior and the Q_w needs to be increased to weight more the measurements.

The value of Q_w is given from the combination of w_1 and w_2 effects; the first is associated to faster dynamics and so it influences more the high-frequency area of the periodogram, while w_2 is going to influence mainly the low-frequency one, because filtered by the integral action of the random walk. The two "knobs" are adjusted until the experimental cumulative periodogram is as close as possible to the ideal one, starting with ρ_2 and afterwards with ρ_1.

After the first batch of trials (Figure 39) the values of ρ_2 that better approximate the ideal behavior are 0.1 and 0.01. In the second batch the choice is further refined: the final configuration is $(\rho_1,\ \rho_2) = (0.1,\ 0.02)$.

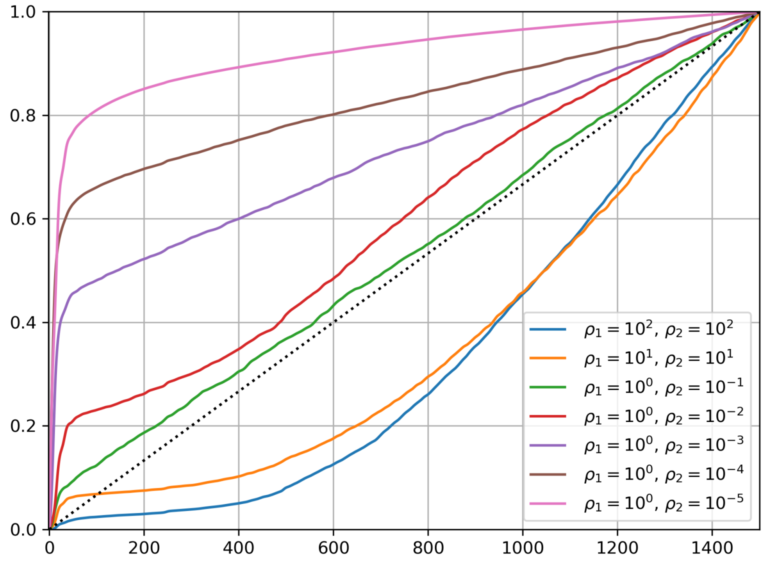


Figure 39: Cumulative periodograms for different choices of ρ_2.

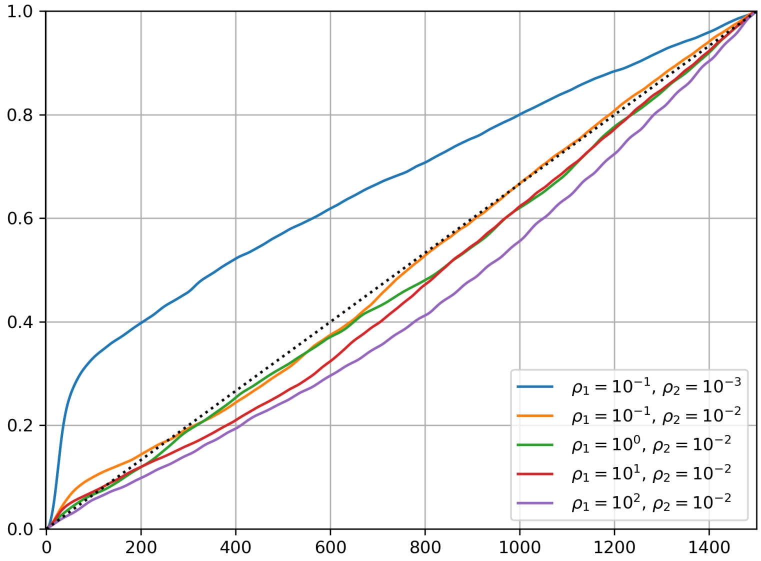


Figure 40: Cumulative periodograms for different choices of ρ_1.

3.4.4 Disturbance Observer Performance

Using the same position controller configuration and disturbance compensation, the Kalman filter performance is compared to the DOB's by evaluating the step response and the constant speed behavior of the system.

Step Response

The system step response behavior is analyzed in different configurations:

- without the external disturbance compensation;
- compensation of the disturbances estimated by the DOB at different bandwidths;
- compensation of the disturbances estimated by the Kalman filter and using the estimated speed in the speed controller;

The results are presented in Figure 41: the collected data doesn't highlight any difference in the position path between the Kalman filter and the others DOB configurations, while the configuration without the disturbance compensation has a slower rise-time and present steady state error caused by the static friction and the absence of the integral action. As the DOB bandwidth increases, the estimated disturbance presents a more impulsive appearance, while the KF's is intermediate between the $300\,\mathrm{rad/s}$ and $600\,\mathrm{rad/s}$ DOBs, indicating that the equivalent KF bandwidth should be intermediate.

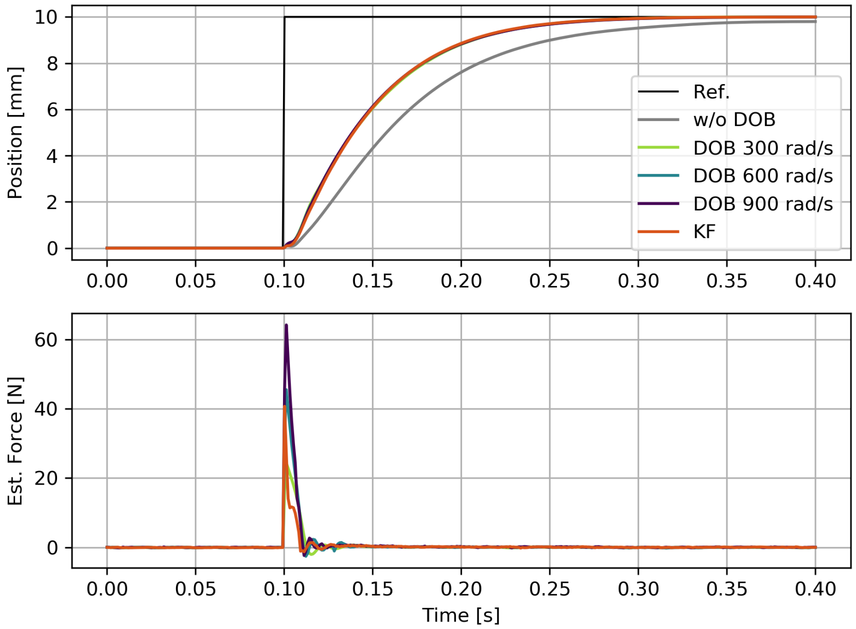


Figure 41: System step response with and without disturbance compensation.

Position Ramp Response

Another insight of the different observers performance can be obtained by choosing as reference a position ramp (constant speed) at $6\,\mathrm{cm/s}$. The results – position error and estimated disturbance force – are shown in Figure 42. As before, the first comment is the huge performance improvement brought by the disturbance compensation, that drastically reduce the position error during the movement. Again, the KF position error performance is intermediate between the DOBs, but this should not surprise, given that we have shown how the two observers structure is equivalent. The KF configuration should however minimize the estimation error and, at first impression, the KF estimate appears "cleaner" than the higher bandwidth DOB configurations.

A zoomed-in version of Figure 42 in presented in Figure 43, where the position error during steady-state (constant speed) can be better analyzed. The KF position error is smaller than the $900\,\mathrm{rad/s}$ and $600\,\mathrm{rad/s}$ DOBs, and very similar to the $300\,\mathrm{rad/s}$. These results prove that increasing the DOB bandwidth improves the system performance up to a certain point: by keep increasing it, the estimation becomes more corrupted by the noise.

From the noise reduction perspective these are great results, but the limited observer bandwidth could affect the perceived responsiveness of the bilateral system, that with the low-bandwidth DOB feels "sluggish" and not natural.

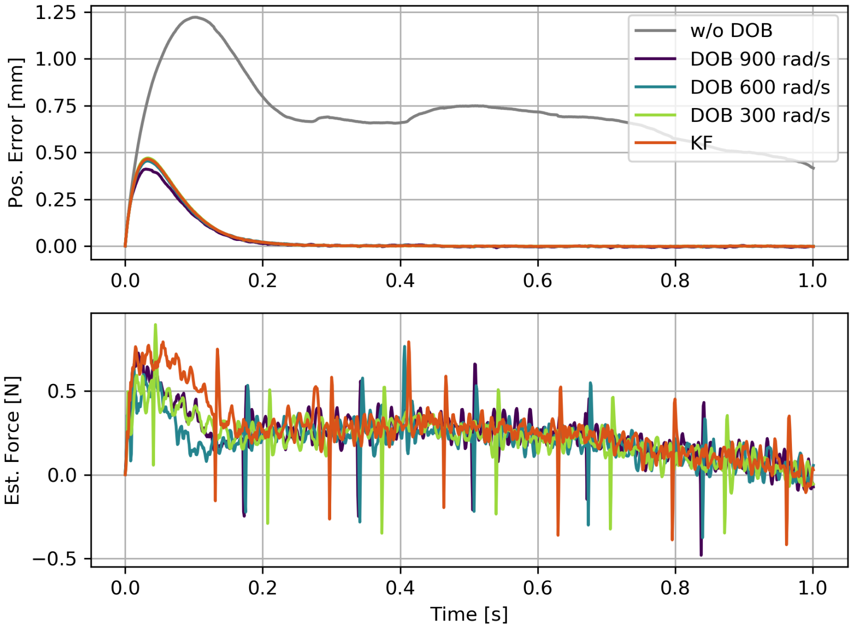


Figure 42: System constant speed with and without disturbance compensation.

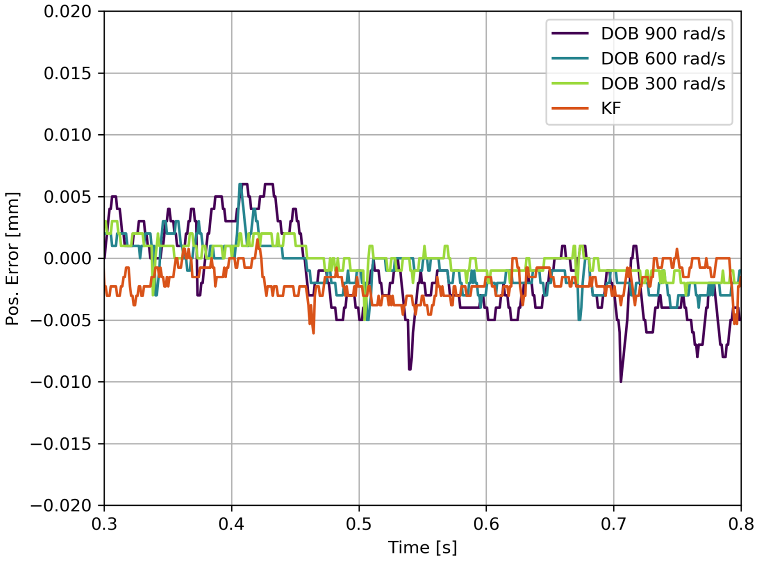


Figure 43: System constant speed with disturbance compensation, particular.

Disturbance Analysis

In all the observer configurations in Figure 42 the estimated external force present a periodic impulsive disturbance, in particular in the three DOB runs these spikes are almost perfectly synchronized. For example, in one run the spikes happen at $t = \{0.172, 0.340, 0.505, 0.671, 0.836\}$ s, with a period of

$$T_s = \frac{0.836 - 0.172}{4} = 0.167\,\text{s}$$

from other tests this phenomenon period appears to be speed-related (the slower the motor, the more sparse the spikes); the motor speed reference is $60\,\text{mm/s}$, obtaining:

$$D_s = 60\,\text{mm/s} \cdot 0.167\,\text{s} = 10.02\,\text{mm}$$

We are looking at an impulsive disturbance that acts periodically every time the motor travels $1\,\text{cm}$. From the motor datasheet we know that the total stator length is $14\,\text{cm}$ and contain 12 windings, each one $1\,\text{cm}$ long, leading to the prospect of a cause-effect relation between the two facts. Is is hypothesized that, when the mover permanent magnets are in a specific position compared to the motor windings, the current/thrust relation degrades significantly and, from the observer standpoint, is equivalent to an impulsive external force.

3.5 EXTENDED FRICTION MODEL

A sinusoidal position/speed reference ($f = 1, 2, 3$ Hz, $A = 1\,\text{cm}$) is used to test the Kalman filter performance. The motor travel was free from any obstacles, therefore the estimated disturbance is imputable to non-nominal values of the thrust constant and mass ($M \neq M_n$, $K_t \neq K_{tn}$), and to the motor friction.

It is hypothesized that the disturbance is caused entirely by the mechanical system static and viscous friction.

For example hypothesizing that $\Delta M = M_n - M = 0.02\,\text{kg}$ (4 times the scale precision) and the maximum acceleration $0.01 \cdot (2\pi f)^2 = 3.55\,\text{m/s}^2$ the inertia term would be:

$$(\Delta M) \cdot (\max \ddot{x}) = (0.02)(3.55) \approx 0.07\,\text{N}$$

while with $K_{tn} - K_t = 33 - 31.8 = 1.2\,\text{N/A}$

$$\frac{K_{tn} - K_t}{K_{tn}} \cdot M \cdot (\max \ddot{x}) \approx 0.08\,\text{N}$$

The motor friction can be estimated from the motor speed using the static/viscous friction model (M1)

$$F = B\dot{x} + F_{st} \operatorname{sign} \dot{x}$$

The results (speed, estimated disturbance and predicted friction and the difference between the two) are reported in Figure 45 for $f = 1\,\text{Hz}$, Figure 46 for

2 Hz and Figure 47 for 3 Hz. Because the friction model has a discontinuity in $\dot{x} = 0$, the estimated friction has a step variation of two times the static friction when the speed changes sign, while the measured force doesn't present this behavior, but instead has a more smooth transition.

The friction model can be extended (M2) to remove the discontinuity by introducing a viscous component of slope K limited between $\pm F_{st}$, as shown in Figure 44. The value $K = 10$ is chosen by trial-and-error, adjusting it until the friction model better approximates the experimental waveforms in the 3 cases.

The extended model improves the friction estimation in all the configuration, but the best results are obtained at the low speeds, where the error between estimated and predicted friction is the smallest.

$$F_{fr} = B\dot{x} + \text{sign}(\dot{x}) \cdot \max\left[K \cdot \text{abs}(\dot{x}); F_{st}\right] \tag{85}$$

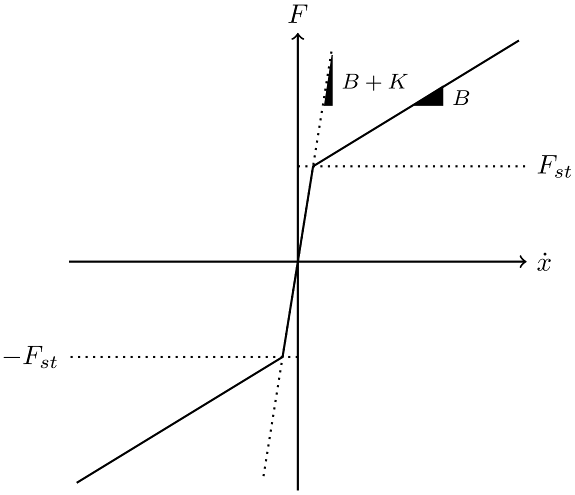


Figure 44: Extended friction model.

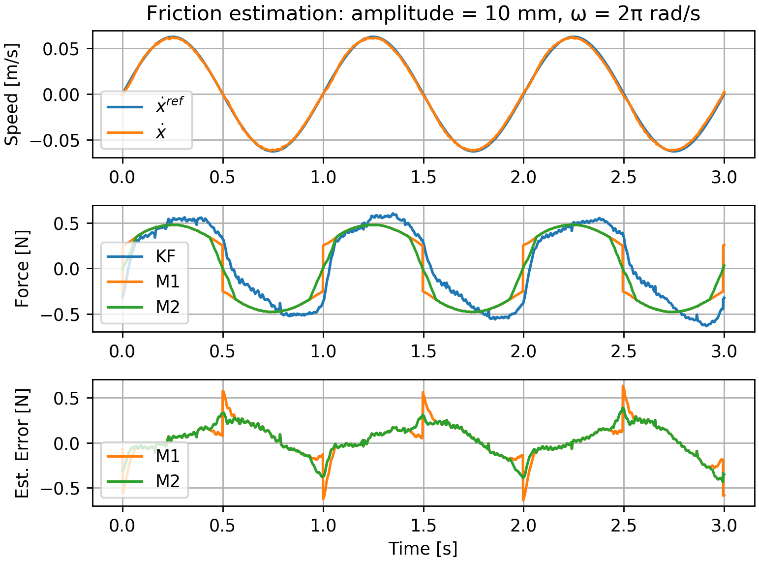


Figure 45: Speed, estimated friction and measured force, error between the two for $f = 1\,\text{Hz}$

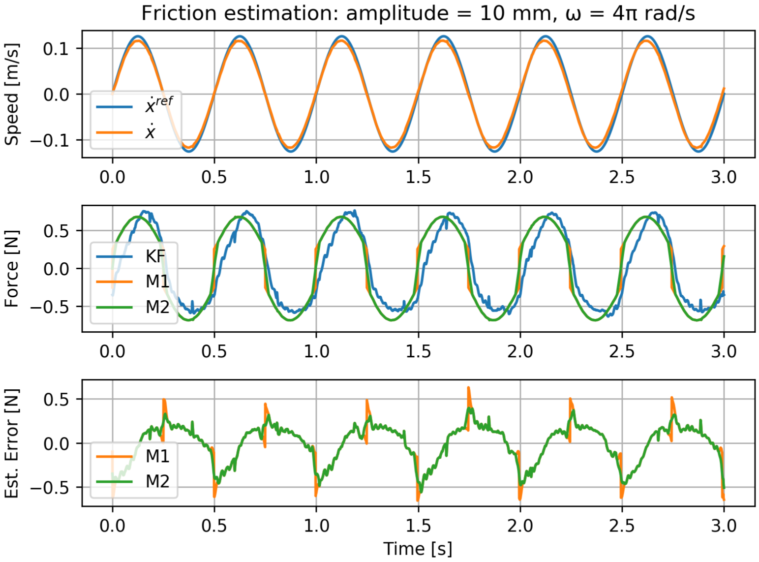


Figure 46: Speed, estimated friction and measured force, error between the two for $f = 2\,\text{Hz}$

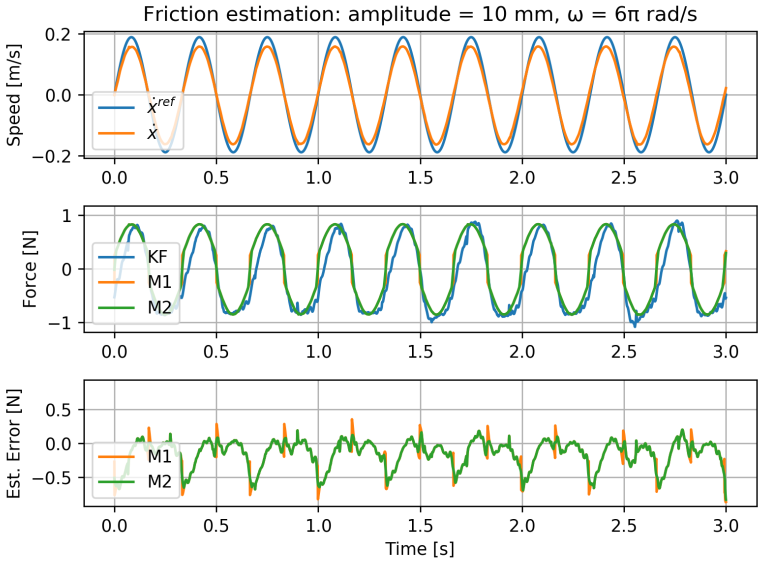


Figure 47: Speed, estimated friction and measured force, error between the two for $f = 3\,\text{Hz}$

4 BILATERAL CONTROL TASK IMITATION

4.1 INTRODUCTION

The bilateral motion copying system described previously permits to save an operator action master references (x_m, $\dot{x}_m$, F_m) to a motion database and reproduce the same action whenever it is needed: the system permits to decouple spatially (thanks to the bilateral control) and temporally (thanks to the motion database) a task execution from the operator. The motion copying system stability was studied by varying the environmental conditions during the task reproduction phase: the bilateral controller behavior is stable even when the environment impedance doesn't match the one used for the task recording, but the reproduced task presents position and force errors. The main drawback of this approach is, in fact, that the saved motion is rigid with respect to environmental changes: for example if the task target position/dimensions or its material (and therefore the stiffness) are changed the saved position and force trajectories don't match with the actual environment impedance, causing an error in the reproduction of the position/force. To cope with these limitations, in this chapter a solution for the imitation of the task is investigated, in particular the following requirements for a suitable task learning algorithms are considered:

- the algorithm should be implementable in the existing controller system
- the algorithm should permit to adapt a saved task to environmental changes, for example the variation of target shape/material/position
- should require as few as training sessions from the operator as possible
- the generated task output should resemble to what an operator would do in the new operating conditions

4.2 THE DYNAMIC MOVEMENT PRIMITIVES FRAMEWORK

The Dynamic Movement Primitive (DMP) [15] was introduced as a framework for humanoid robot control and trajectory planning, inspired by the *motor pattern generators* studies in neurobiology, emulating how biological neural circuits generate the motion signals. The DMP output **y**, used as the virtual master for the bilateral controller, is generated by a dynamic system with known stability properties and point attraction behavior to mimic the operator action. The use of a dynamic system as the trajectory generator permits to have fine control over the properties of the output reference signal. The DMP dynamic system is modulated by a non-linear forcing term **h** needed to memorize/model a specific operator task. A DMP model is used to learn the trajectories associated with a specific operator action; in the case of a bilateral control task, for each system degree of freedom, both the position and the force reference signals need to be included. For the 1-DOF mechanical test bench the system reference is composed by the bilateral master-side position x_m and force F_m references. A particular action/DMP model is characterized by:

- t_D is the action duration
- $\tau > 0$ is the action time scaling, that can be used to slow down or speed up the task reproduction speed;
- $g^p = x_m(t_D)$ is the action scaling factor, it permits to scale the position reference to adapt the saved task to environmental changes;
- g^f is the force scaling factor, it permits to scale the force reference to adapt the saved task to the action target changes;
- a vector of N weights $[w_0, \dots, w_{N-1}]$ associated to N radial basis functions to store each action trajectory.

The original [15] DMP model has been improved and simplified in [16]: each component of the dynamic movement primitive framework is presented below.

Canonical system

The canonical system is an autonomous system which output is used as a phase or as a clock by the other DMP components during the task reproduction. Both 1-time discrete and rhythmic movement can be generated by the canonical system to match the task type:

$$\text{Single} \qquad \dot{z} = -\frac{1}{\tau\, t_D} z \tag{86}$$

$$\text{Rhythmic} \qquad \dot{z} = \tau\, \omega \tag{87}$$

where ω is the angular speed that control the task reproduction speed and t_D is the action duration (in seconds). In the present work only non-repetitive task are considered, but the same framework used has been verified for both [15, 17]. The canonical system evolution is then:

$$z(t) = \exp\left[-\frac{t}{\tau\, t_D}\right] \tag{88}$$

The DMP task trajectories are modeled using the time coordinate z that decouples the task reproduction time from the control system time. By using a common canonical system z:

- the time scaling of a task is obtained by simply changing the coefficient τ
- it is ensured that the force and position signals of one or more DOF are synchronized under τ changes

Dynamic System

As presented before, a dynamic system is used to generate the system reference trajectories for the bilateral controller. The dynamic system output is modulated by the task end-state goal **g** and by the forcing term **h**: the first provides the point-attraction behavior, the second the non-linear evolution that permits to learn the operator movements. In [15] a simple second-order linear system is introduced to "emulate" the operator dynamics and it is presented below.

$$\ddot{\mathbf{y}} = \alpha\left(\beta\left(\mathbf{g} - \mathbf{y}\right) - \dot{\mathbf{y}}\right) + \mathbf{h}$$

The system output **y** is used as the bilateral control virtual master, while the action goals **g** can adapt the saved task to new environmental conditions. Finally, to add the capability to slow down or speed up the task reproduction the temporal scaling factor τ is included:

$$\tau^2\, \ddot{\mathbf{y}} = \alpha\left(\beta\left(\mathbf{g} - \mathbf{y}\right) - \tau\, \dot{\mathbf{y}}\right) + \mathbf{h} \tag{89}$$

Forcing Term

The forcing terms $\mathbf{h} = [h^p,\ h^f]$ store the non-linear behavior of the action position and force trajectories. The forcing term structure should be chosen so that it can represent complex patterns and that can be also easily modeled after the reference trajectories. In [15], a linear combination of N radial basis functions $\psi_i(z)$ for each DOF j, with $i = 0, 1, 2, \ldots, N-1$, is used:

$$h(t) = \frac{\sum_{i=0}^{N-1} \psi_i(z(t))\, w_i}{\sum_{i=0}^{N-1} \psi_i(z(t))} \left(g - y(0)\right) z(t) \tag{90}$$

Because we need to model both the master position and force reference, two set of weights and two goals are saved to generate the $h^p,\ h^f$ references. The $i = 0, 1, 2 \ldots, N-1$ Gaussian radial basis functions $\psi_i(z(t))$ are centered at the canonical system times $c_i,\ i = 0, 1, \ldots, N-1$ equally spaced along the Canonical System evolution, and have widths σ_i [16, 4]:

$$\psi_i(t) = \exp\left[-\frac{(z(t) - c_i)^2}{2\sigma_i^2}\right] \tag{91}$$

$$\sigma_i^2 = \frac{c_i}{N^2} \tag{92}$$

The Gaussian radial basis functions ψ_i are equally distributed along the task duration t_D: $\tau\ t_D$ is divided into N time windows, each of those starts at

$$d_i = \frac{\tau\, t_D}{N} \cdot i$$

that, in the $z(t)$ coordinate, becomes

$$c_i = \exp\left[-\frac{d_i}{\tau\, t_D}\right] = \exp\left[-\frac{\tau\, t_D\ i}{\tau\, t_D\ N}\right]$$

and can be simplified to

$$c_i = \exp\left[-\frac{i}{N}\right] \tag{93}$$

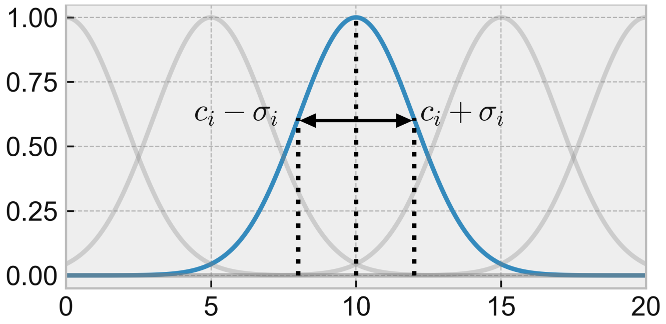


Figure 48: Example of the a Gaussian radial basis function.

The forcing term in (90) is modulated by $(\mathbf{g} - \mathbf{y}(0))$, ensuring that is properly scaled under a change of movement/force amplitude, obtained by changing the task position and force goals g^p, g^f. As the canonical system transitions, it activates the radial basis functions that, weighted by w_i, generate the forcing term for each of the dynamic system outputs. In Figure 49 the activations of 7 radial basis functions and their weighted sum, as described by (90), is presented.

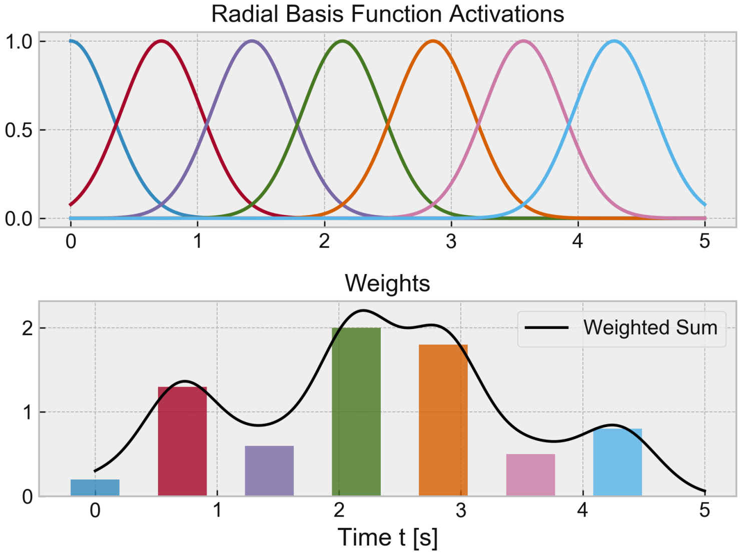


Figure 49: RBF activations and the forcing term output, calculated from the weighted sum of the activation functions.

At each action start the system state $\mathbf{y}(0)$ is attracted to the goal $\mathbf{g}$ by the $(\mathbf{g} - \mathbf{y})$ term, whereas $\mathbf{h}$ provide the non-linear behavior and τ provides the temporal scaling properties. The (89) can be rewritten in the Laplace domain, where the behavior of the dynamic system is more explicit:

$$Y(s) = \frac{\alpha\beta}{\tau^2 s^2 + \alpha\tau s + \alpha\beta}\left[G + \frac{H(s)}{\alpha\beta}\right] \tag{94}$$

From the (94) it can be recognized that the dynamic system acts like a second-order low pass filter, that is tuned to present critical damping:

$$\beta = \frac{\alpha}{4} \quad \Longleftrightarrow \quad \xi = \frac{\alpha\tau}{2\sqrt{\tau^2\alpha\beta}} = 1 \tag{95}$$

with cutoff frequency $w_n = \alpha/\tau$.

Dynamic System Bandwidth

The dynamic system cutoff frequency $w_n = \alpha/\tau$ is chosen with the assumption that the mechanical system is handled by the operator hand and therefore it should have a similar behavior. Various studies regarding the dynamic properties of the hand movements are present in the scientific literature, of which several have tried to estimate the "control bandwidth" of the operator. In [18], the subjects are asked to balance an inverted pendulum with a decreasing length L: the pendulum dynamics are governed by an unstable pole in $s = \sqrt{g/L}$ rad/s and so, by reducing the pendulum length (therefore increasing the pole frequency), the bandwidth needed to stabilize the system increases. From the experimental results, the estimated human control bandwidth is limited to frequencies around 3 Hz while higher frequencies involve a loss in the movements amplitude precision. Similar results are presented in [19], where it is shown that the precise amplitude of finger movement can be controlled up until $3-4$ Hz. The DMP dynamic system is then tuned to have a bandwidth of 4 Hz, or:

$$w_n = 2\pi \cdot 4\,\mathrm{Hz} \approx 25\,rad/s \tag{96}$$

obtaining, with $\tau_0 = 1$,

$$\begin{aligned} \alpha &= 25 \\ \beta &= \alpha/4 = 6.25 \end{aligned}$$

The same τ term is present in both the canonical and dynamic systems: when the action is speed up or down, the filter bandwidth is changed to preserve the output trajectory path, otherwise the high frequency components will be attenuated by the second order low-pass filter.

Fitting the DMP Action Weights

A DMP model can "learn" a task trajectory reference by fitting a set of parameters (weights) of the non-linear functions h^p, h^f to imitate the recorded operator task. In the case of multi-DOF system, a set of parameters are associated to each DOF to realize complex movements. DMPs have been successfully used to learn the *Ball-in-a-Cup* game with a 7-DOF robotic manipulator [17]. For each action parameter i we define a quadratic loss function J_i that measures how well each RBF approximate the reference signal. The learning objective is then finding each weight w_i^p, w_i^f that minimize the weighted quadratic loss function

$$J_i = \sum_{k=0}^{M-1} \psi_i(z(k)) \left[h^{ref}(k) - \gamma(k) w_i\right]^2 \tag{97}$$

where $M = D/T$ is the number of task samples, $\gamma(k) = [g - y(0)]z(k)$ and $h^{\,ref}(k)$ the forcing term we want to learn. $z(k)$ indicates the sampled version of $z(t)$, $z(k) = z(kT)$. To solve the optimization problem, the loss function can be rewritten in the vectorized form

$$J_i = \left[\mathbf{h}^{\,ref} - \mathbf{s}\,w_i\right]^T \Psi_i \left[\mathbf{h}^{\,ref} - \mathbf{s}\,w_i\right] \tag{98}$$

where $\mathbf{h}^{\,ref}$ is a vector of the h_k, $k \in \{0, 1, \dots, M-1\}$ samples, Ψ a diagonal matrix of the radial basis functions activation,

$$\Psi_i = \begin{bmatrix} \psi_i(0) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \psi_i(M-1) \end{bmatrix} \in \mathbb{R}^{M\times M}$$

and $\mathbf{s}$ a vector of the $\gamma(k) = g_j\ z(k)$ terms. The solution of the locally-weighted linear regression is

$$w_i = \left(\mathbf{s}^T\Psi\mathbf{s}\right)^{-1} \mathbf{s}^T\Psi\,\mathbf{h}^{\,ref} \tag{99}$$

The position forcing term reference $h^{p,ref}$ is calculated rearranging the (89)

$$\mathbf{h}^{p,ref} = \alpha\left(\beta\left(\mathbf{y}^{p,ref} - \mathbf{g}^p\right) + \tau\,\dot{\mathbf{y}}^{p,ref}\right) + \tau^2\,\ddot{\mathbf{y}}^{p,ref} \tag{100}$$

while for the force's

$$\mathbf{h}^{f,ref} = \alpha\beta\left(\mathbf{y}^{f,ref} - \mathbf{g}^f\right) \tag{101}$$

where $\mathbf{y}^{ref}$ is the vector of the master position ($\mathbf{y}^p = x_m$) and force ($\mathbf{y}^f = F_m$) reference.

4.3 BILATERAL ACTION DMP MODEL

4.3.1 Action Saving

The presented Dynamic Movement Primitive framework is used to save an operator action. Multiple actions can be combined to generate a new task and by adjusting the DMP parameters it is possible to deal with environmental changes. The forces estimated by the DOB include all disturbances acting on the motor, such as static and viscous friction, but we are interested in saving a generalized version of the operator task, invariant to time and amplitude scaling. For this reason, estimated external disturbances are subtracted from the force signal before the parameters fitting, using the friction model presented in section 3.5. In particular the static $\hat{F}_{st}$ and viscous $\hat{B}$ friction coefficients have been estimated in subsection 2.6.2 and $K = 10$; the *net* external force F_n is calculated as:

$$\hat{F}_n = \hat{F} - F_{fr} = \hat{F} - B\dot{x} - \mathrm{sign}(\dot{x}) \cdot \max\left[K \cdot \mathrm{abs}(\dot{x}); F_{st}\right] \tag{102}$$

4.3.2 Action Loading

Having recorded various operator tasks, it is possible to combine them into a single task, possibly changing the time constants τ to speed it up or down and the goal g to scale the position or force references. The generated references are joined together by an interpolator that ensures continuous position and speed references by joining the different actions using a 3rd-degree polynomial for the position and a 1st degree (linear) for the force. Finally, the net force reference F_n is supplemented with the estimated friction force for its feed-forward compensation.

4.3.3 Number of RBFs

When an action position and force references are used to model the Dynamic Movement Primitives forcing terms h^p, h^f, the number of radial basis functions N needs to be chosen. In the machine learning jargon, N is defined as an *hyperparameter*, a user-defined parameter that control the model complexity and its accuracy. Increasing N means that, during equal time intervals, the higher RBFs density can better approximate the position/force signals to obtain an higher quality representation of the original trajectory, at the cost of a higher computational load. As an example, an arbitrary signal h^{ref}, generated from the combination of two sinusoidal waveforms at 3 and 6 Hz, is used as the reference for a possible forcing term signal:

$$h^{ref} = \sin(3 \cdot 2\pi t) + \sin(6 \cdot 2\pi t)$$

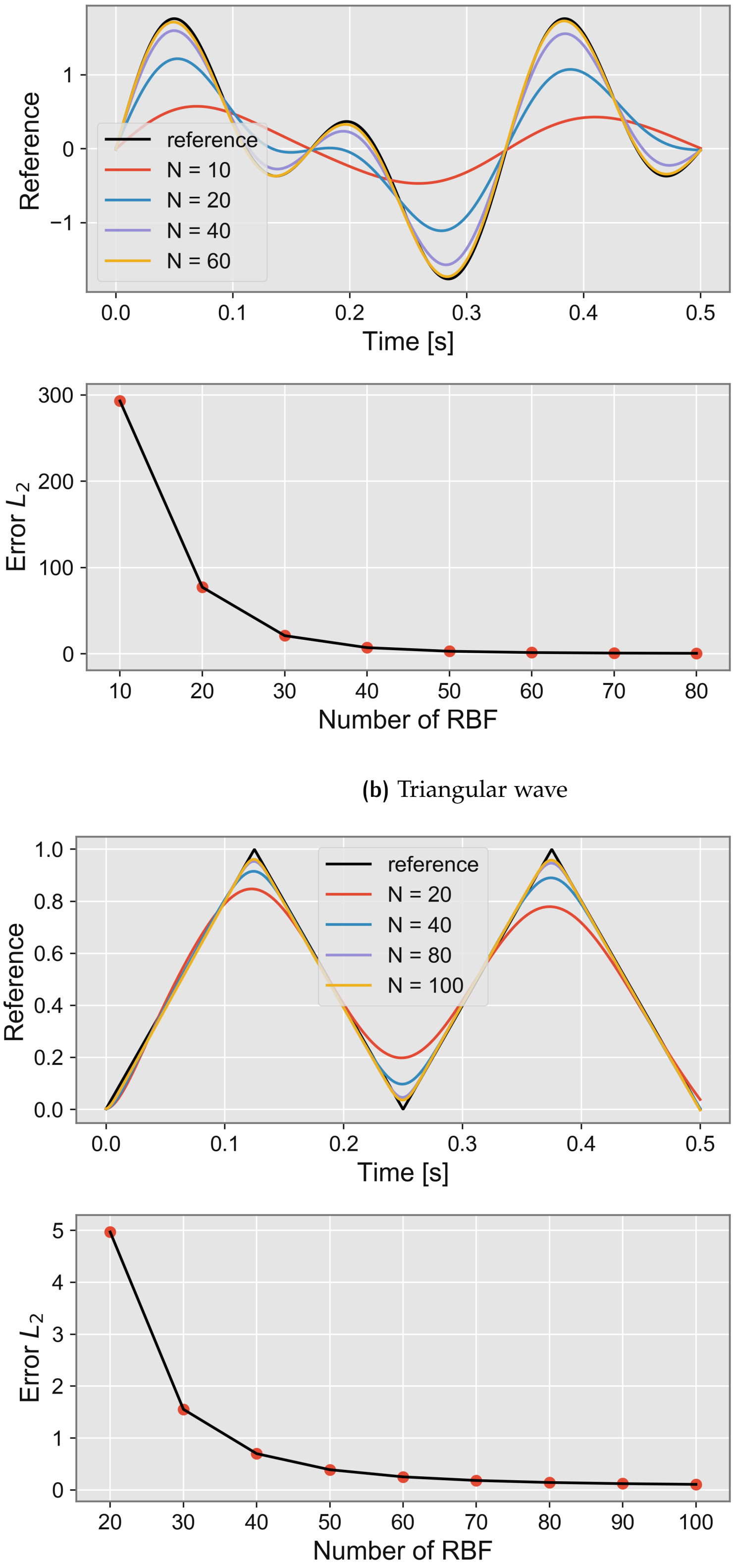
The signal is approximated using different values of N and the outputs are displayed in Figure 51a. As the number of Gaussian radial basis functions N increases from 10 to 60, the reference signal is better approximated by the weighted sum of RBFs, as it is proved by the quadratic loss index L_2, defined as

$$L_2 = \sum_{k=0}^{M} \left[h^{ref}(k) - h(k)\right]^2$$

Another example is presented in Figure 51b, where a triangle wave is used as the reference h^{ref}. As expected the quadratic error decreases as the number of RBF increases, better approximating the reference. The number of RBF needed to approximate the function to a given precision (defined as the maximum absolute error) depends on the reference signal characteristics. Given the smooth kernel shape of the Gaussian RBF, even with a large number of elements it is difficult to approximate non-differentiable points like the cusp in the triangular wave. As presented above, the human hand movements have a limited control bandwidth, an so this limitation shouldn't represent a problem for the action modeling:

- the position reference doesn't present non-derivable points like the triangular wave did;
- the force reference, on the other hand, includes high frequency components that won't be present in the DMP model and so are lost. Of those high frequency components:
  - some are estimation noise, therefore the filtering of those is a positive side effect of the model-fitting process;
  - the remaining fraction is actual lost information.

Figure 50: Approximation of different signals using N Gaussian radial basis functions



(a) Combination of two sinusoidal waves at 3 and 6 Hz

(b) Triangular wave

In practice the number N of RBF is incremented by 10 until the position approximation error is reduced to $0.2\,\mathrm{mm}$ and the force error to $0.5\,\mathrm{N}$. For the majority of the actions, the limiting condition is the one on the force.

4.3.4 Computation time

The number of Gaussian radial basis functions is a trade-off between the forcing term approximation precision we want to achieve, the computational load necessary to fit the corresponding weights and the memory space needed to archive the weights. The fitting of the DMP model for a particular operator action happens after the bilateral task is completed, therefore during offline operation. In Figure 52 the time needed to fit a test action reference (the triangular wave shown in the previous section) is presented. Because the algorithm runs as a program on a general purpose PC where other programs are active, to obtain more significant results each calculation duration is obtained as the mean of 10 algorithm runs. From the collected data, it emerges that the computation time increases linearly with the number of RFB, in particular it takes a base time of approximately $13\,\mathrm{ms}$ for 1 RBF and $0.065\,\mathrm{ms}$ for each additional RBF on an Intel i7 at $2.8\,\mathrm{GHz}$.

$$t_{calc} = N_{rbf} \cdot 0.065\ \mathrm{ms} + 13\,\mathrm{ms}$$

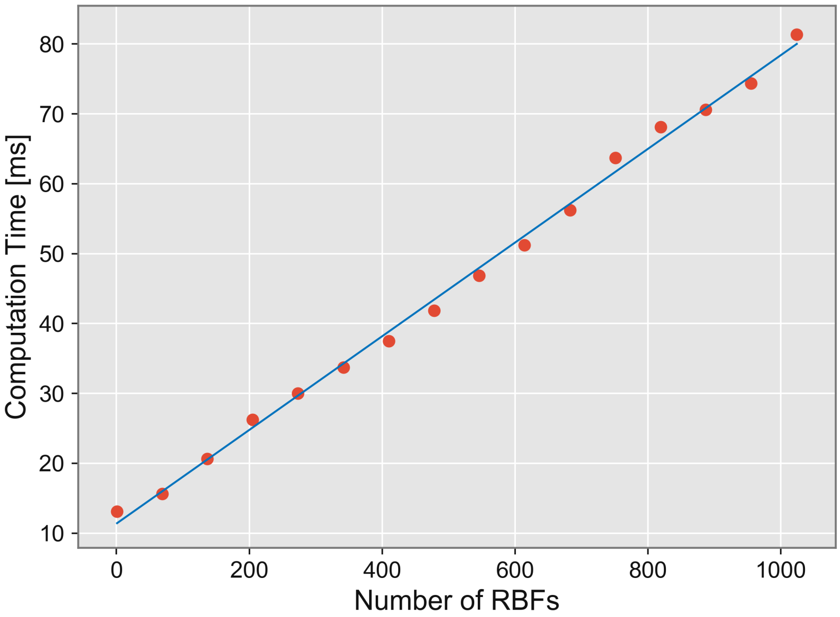


Figure 52: Computation time for different numbers of RBFs.

4.3.5 Force Scaling

To adapt a saved action to an environment different from the one present during the recording, the action goals g_p, g_f can be modified. By scaling the two parameters proportionally the applied force is proportional to the position displacement, that is compatible with the interaction of objects with linear stiffness and for actions that start at the contact point. As it will be presented later, different behaviors can be achieved tweaking the position and force goals independently, useful for example when the stiffness of the target changes. One limitation of the movement scaling approach is that, if the starting value coincides with the goal, the term $[g - y(0)] \simeq 0$ deteriorates the forcing term dynamics (90). A possible solution consists in splitting the task into two sub-tasks or, if the task is symmetric with respect to a point in time, consider only the first half and generating the other half using $g = -1$.

4.4 TASK IMITATION

From the experimental bench bilateral control setup two primitive actions are isolated:

1. the movement to position the end-effector in contact with the target (a sponge) from the home position;
2. the interaction with the environment, by pressing the end-effector against the sponge

The two actions position and force references are used to model two DMPs, Figure 53 and Figure 54.

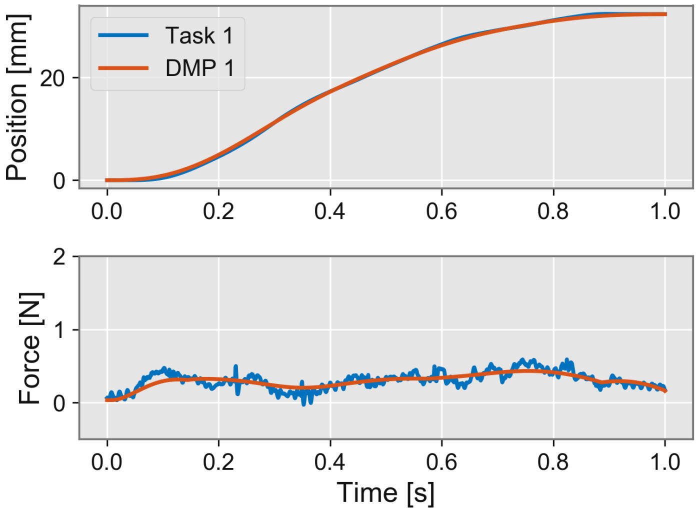


Figure 53: Action Primitive 1 – Simple motor movement, recorded samples (blue) and DMP task model (orange).

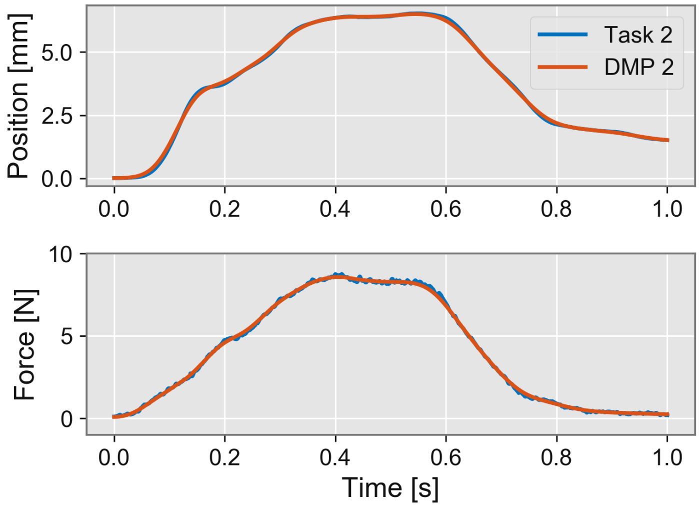


Figure 54: Action Primitive 2 – Interaction with the environment (sponge), recorded samples (blue) and DMP task model (orange).

4.4.1 Bilateral task scaling

After the task have been modelled, the scaling capabilities of the DMP framework applied to the bilateral control are tested.

Time Scaling

The action execution can be sped up or down adjusting τ: in Figure 55 the task time constant τ is doubled and halved to generate the references (black dotted lines) of other two actions. The references are then used as the virtual master references for the motion copying system. The collected experimental results are reported in Figure 55: for both actions the task reproduction output matches the generated references; while the estimated force is faithfully reproduced, there is a significant position error ($0.43\,\mathrm{mm}$) compared to the bilateral control operation.

Amplitude Scaling

The bilateral action can be also amplitude scaled: this operation is "trickier" than the time scaling because the two references need to match the environment impedance. A simplified environment model is used, where it is hypothesized that the damping is negligible in respect to the (constant) stiffness K

$$F = K\,\Delta x \tag{103}$$

After these considerations, if the position and force references are changed proportionally, the original action can be scaled accordingly. The same action used in the time scaling is modified by doubling and halving the position and force goals g^p, g^f; the results are reported in Figure 56.

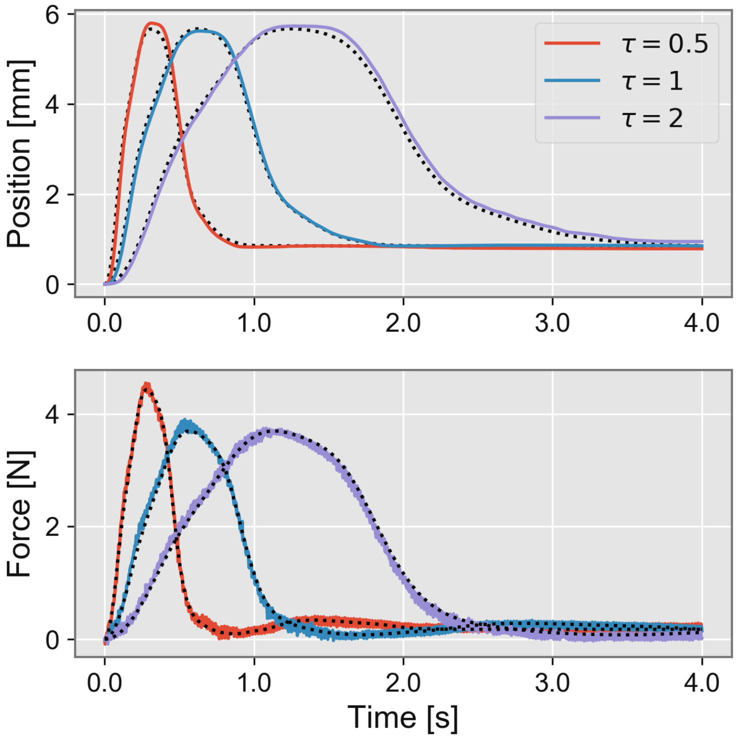


Figure 55: DMP model temporal scaling capabilities

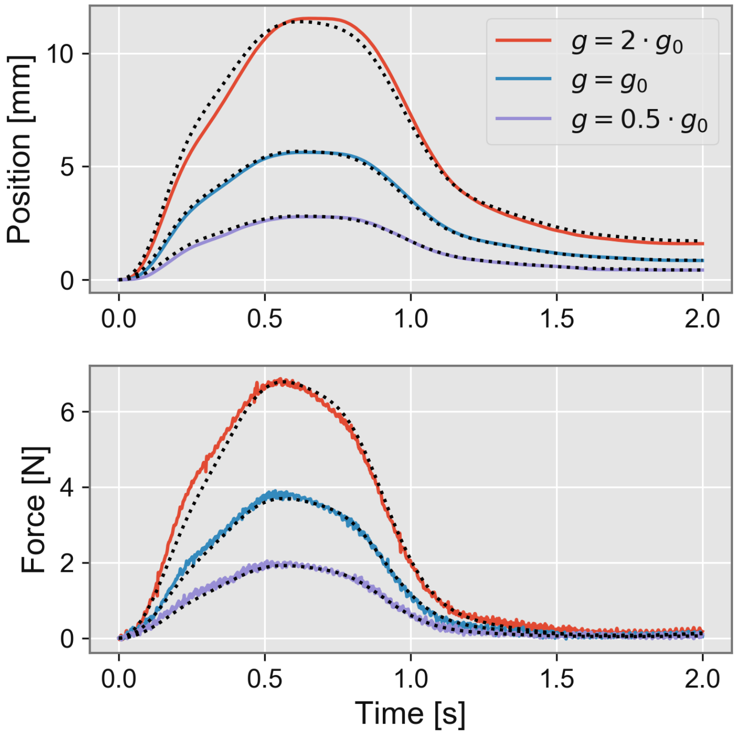


Figure 56: DMP model position/force scaling capabilities

4.4.2 Task generation

The saved bilateral control action primitives can now be used to generate a new complete task without the need to record new operator actions. The two modeled primitives (Figure 53, Figure 54) are used to generate a complete task as indicated in Table 9. The generated position and force references, that will be loaded by the motion copying system and are already supplemented with the estimated friction, are presented in Figure 57 as well as the result trajectory obtained by the bilateral controller. From the experimental results, the proposed framework and simplifications are supported by the experimental results, that, albeit not perfectly matching, show a good position tracking ($\max |e_p| = 1.1\,\mathrm{mm}$) as well as force mirroring, meaning that the reference position/force relationship successfully approximates the experimental conditions.

Table 9: Task generation reference actions.

Sector	Action	g^p	g^f	τ
A	Action 1	g^p_0	g^f_0	0.9
B	Action 2	g^p_0	g^f_0	1
C	Action 2	g^p_0	g^f_0	2/3
D	Action 2	$1.4g^p_0$	$1.4g^f_0$	1
E	Action 1	$-g^p_0$	$-g^f_0$	0.5

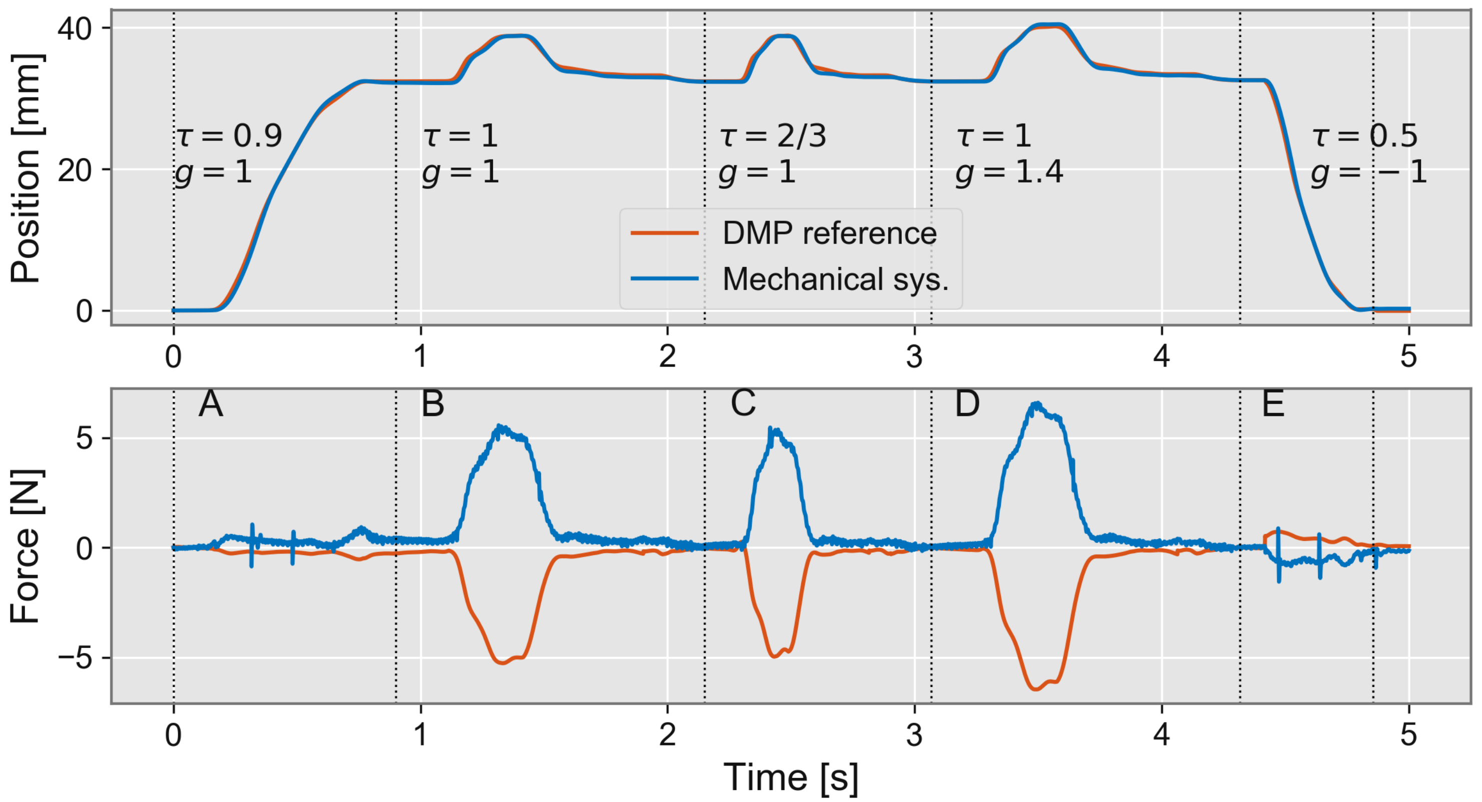


Figure 57: Position and force reference and output of the bilateral control, reference (orange) generated by the combination of DMP actions and mechanical system output (blue).

4.5 TASK ADAPTATION TO TARGET CHANGES

In the previous section the recorded bilateral control primitives are used to model an operator action and are combined to realize a new task, with the hypothesis that the environmental conditions remain the same as during the primitives recording by the operator. In this section it is studied the problem of adapting recorded primitives to environmental changes. Two possible conditions are considered: the target position/shape or its material (more specifically its stiffness) are changed. To asses the adaptation capabilities a test task is chosen (Figure 58, Table 10), composed of two actions, built similarly to what was done in the previous section. A performance baseline was obtained by changing the environmental conditions (target material and position) and executing the saved task as it is, Figure 59.

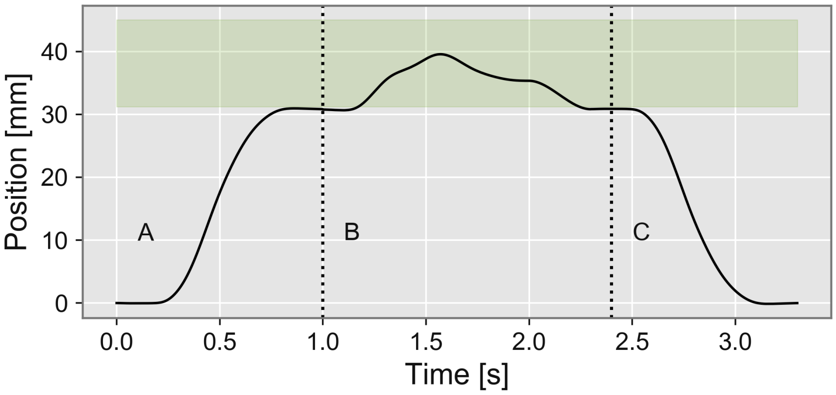


Figure 58: Original target position (green box) and master position reference.

Sector	Action	Duration	τ	Note
A	Action 1	1 s	1	Movement from the home position to contact point
B	Action 2	1.4 s	1	Interaction with the environment
C	Action 1	1 s	1	Return to the home position

Table 10: Default task configuration

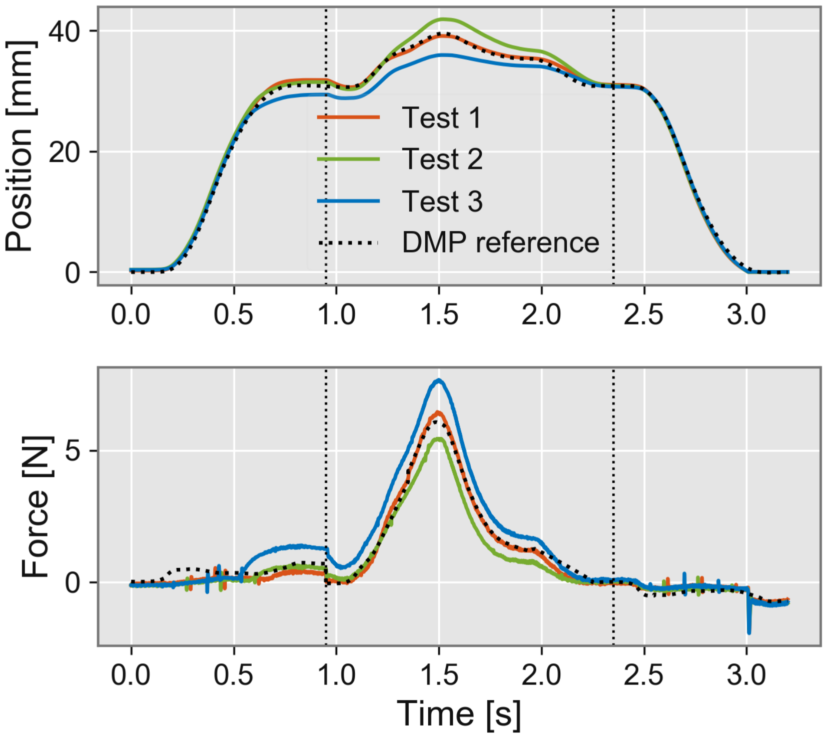


Figure 59: Test task in presence of environmental changes – Test 1 with the default configuration, Test 2 with a softer target, Test 3 with a stiffer target positioned closer to the home position.

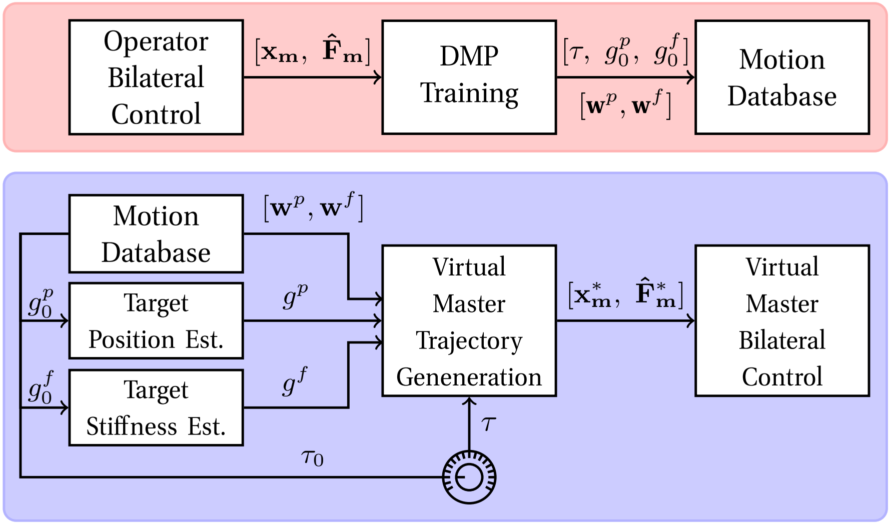

Operator Bilateral Control
$[\mathbf{x}_\mathrm{m},\ \hat{\mathbf{F}}_\mathrm{m}]$
DMP Training
$[\tau,\ g_0^p,\ g_0^f]$
$[\mathbf{w}^p, \mathbf{w}^f]$
Motion Database
Motion Database
$[\mathbf{w}^p, \mathbf{w}^f]$
g_0^p
Target Position Est.
g^p
g_0^f
Target Stiffness Est.
g^f
Virtual Master Trajectory Geneneration
$[\mathbf{x}^*_\mathrm{m},\ \hat{\mathbf{F}}^*_\mathrm{m}]$
Virtual Master Bilateral Control
τ
τ_0

Figure 60: Task recording and environmental task adaptation logic blocks.

Strategy for the Action Primitives Adaptation

In Figure 60 the proposed strategy to adapt the saved bilateral primitives is presented as a block diagram:

- in the first phase (red background) the operator action is modeled and saved to the motion database
- in the second phase the action is loaded from the database and, from the new environment information collected during an exploration procedure, adapted to the new conditions.

Said exploration procedure is presented in the next pages.

4.5.1 Target Position Estimation

The first step to adapt the action primitive to the new target position is the estimation of said position. To do so the following steps are performed:

1. the linear motor is accelerated from the home position to the constant speed v in a time $t_1 = 0.1\,\mathrm{s}$ (constant acceleration of $a = v/t_1$);
2. after $t_2 = 0.2\,\mathrm{s}$ the estimated external force is monitored and a force threshold F_t is activated;
3. once n consecutive samples exceed the force threshold, the end-effector is considered in contact with the target and the motor output is disabled – this option permits to avoid false-positives at the cost of a delay in the contact detection
4. being the end-effector in contact with the target, the position is registered after the motor speed drops to zero, to allow for possible overshoots to even out.

The choice of the v, F_t, n values is a trade-off between the speed of estimation, maximum impressed force and the detection false-positives tolerance. It is arbitrarily set $F_t = 0.5\,\mathrm{N}$ to limit the maximum force exerted on the target. With $n = 5$ no false-positive detections have occurred during the experimental testing, while for example $n = 2$ produced several. For the choice of the motor speed different tests have been conducted, with $v = 1,\ 2,\ 4\ \mathrm{cm/s}$. The experimental results for different values of v are reported with a steel target in Figure 61 and with a soft target in Figure 62. The test with $v = 4\,\mathrm{cm/s}$ in Figure 61 (red lines) force estimation presents disturbances that exceed the force threshold around $t = 0.25\,\mathrm{s}$, but lasts for only 3 consecutive samples and therefore it isn't detected as a contact, avoiding the false-positive detection described before.

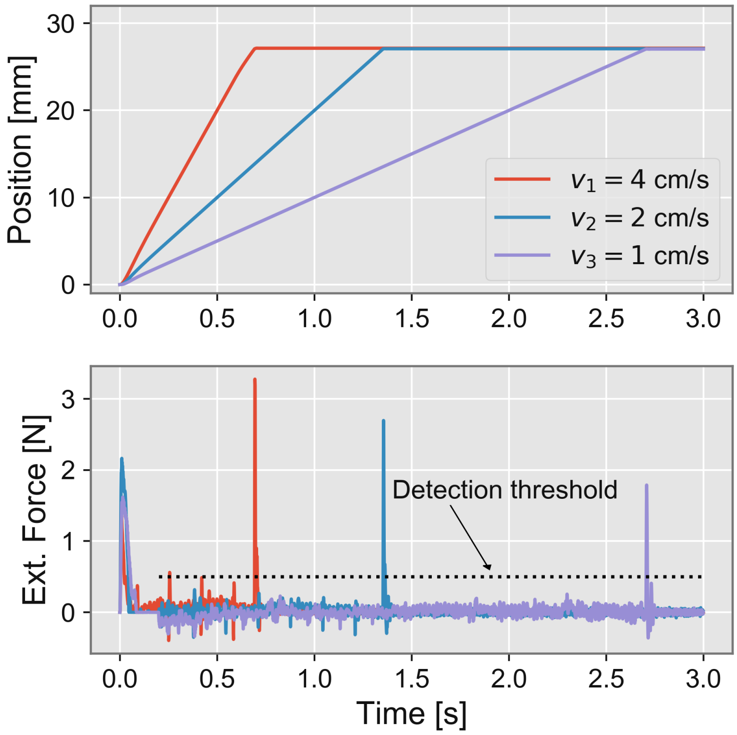


Figure 61: Target position estimation - Hard (steel) target.

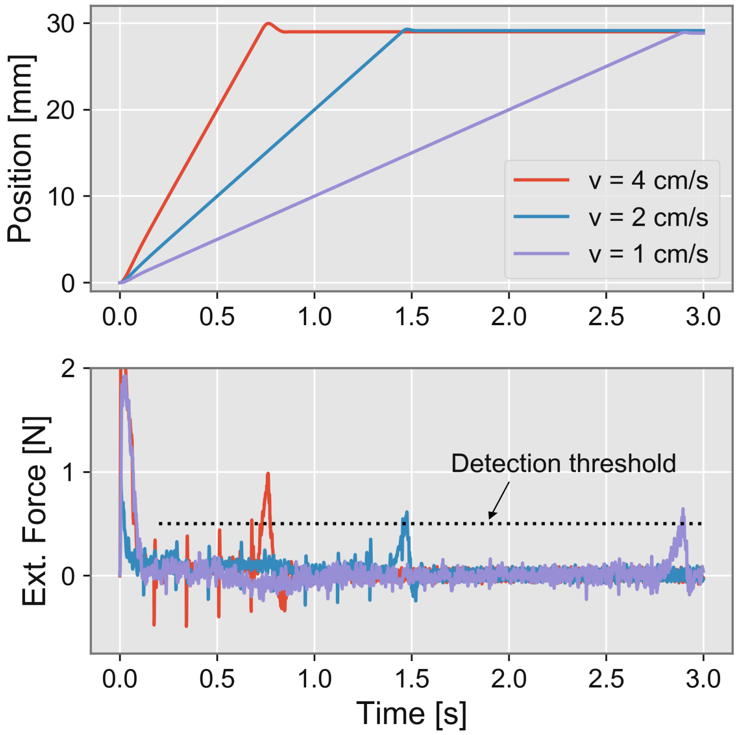


Figure 62: Target position estimation - Soft (sponge) target.

Other than the estimation total duration, the effects of the different motor speeds are:

- for the hard target the only difference is the impact force
- for the soft target an overshoot is caused by the motor inertia

In Table 11 are reported the results from the soft target tests: overshoot, time of the initial contact and the total estimation duration (until the end-effector is resting still at the target edge, therefore including the overshoot dynamics).

Speed [cm/s]	First contact [s]	Total duration [s]	Overshoot [mm]
1	0.731	0.844	1.21
2	1.467	1.527	0.23
4	2.887	2.889	0.09

Table 11: Position detection results – Soft target

4.5.2 Target Position Modification

The position scaling properties and the target position detection procedure are then tested together to evaluate the target position adaptation capability of the proposed framework. The same target used during the action recording is moved to two different position (closer and further from the original point). After the estimation of the target position x_t^*, the position goal g^p of the first ($0 \to x_t^*$) and third action ($x_t^* \to 0$) is updated to match the new value.

The experimental results are presented in Figure 63. While the master force reference matches very closely the recorded ones, the position tracking presents some flaws during the action reproduction, especially during the contact with the target.

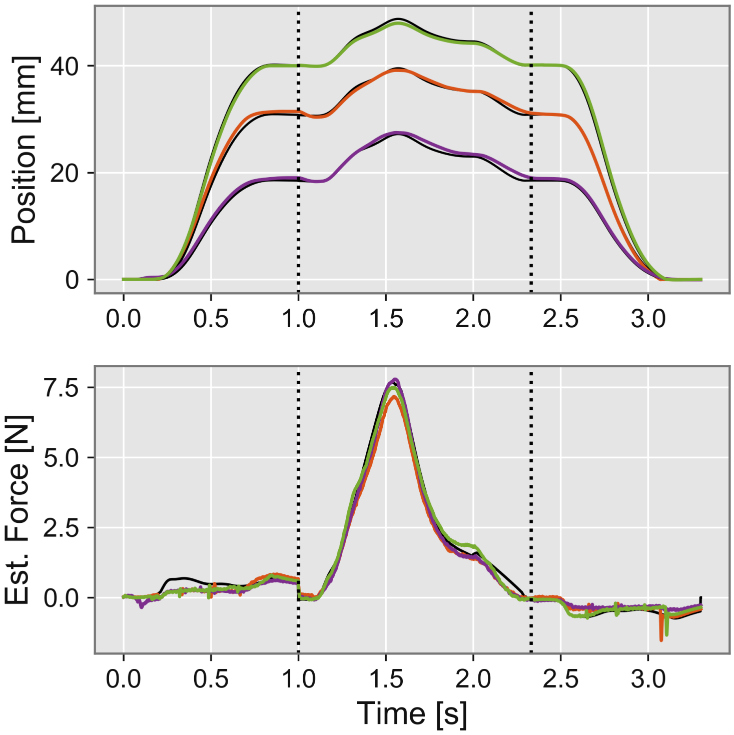


Figure 63: Task adaptation to targets at different positions from the home point – Original task (orange), further (green) and closer target (purple).

4.5.3 Target Stiffness Estimation

The second option to adapt an action primitive to a new environment is to change the position and force references independently to match the new target stiffness (i.e. made by a different material). The target stiffness K is hypothesized constant and is estimated by comparing the displacement Δx with the estimated reaction force F during the environment interaction.

$$F = K\,\Delta x \tag{104}$$

The stiffness estimation is performed after the target position detection, when the end-effector is still resting against the target edge. Two strategies can be followed:

- perform a position control and record the estimated force
- perform a force control and record the end-effector position

Both have some advantages and limitations. The position control has a stable behavior, but drives the motor ignoring the environment reaction forces; this is not a problem for sturdy objects, but it could be for fragile ones. A pure force control permits to have direct control over the applied forces, but on the other hand it is not stable: in case the target object moves from the motor trajectory, the force control accelerates the motor until the reference force is reached, but this condition could be against an obstacle or a mechanical stop.

Position Control

For the target stiffness estimation using the position control the following position/speed signals are used as the bilateral control master reference, while the force control is temporarily disabled by setting the force gain to zero, $K_f = 0$. The displacement Δx, estimated force and the linear regression results (K_1 original target, K_2 stiff one, K_3 soft one) are reported in Figure 64.

$$\begin{aligned} x_m^* &= 0.02 \sin(0.2\pi t) \\ \dot{x}_m^* &= 0.2\pi \cdot 0.02 \cos(0.2\pi t) \\ t &\in [0,\ 5]\ s \end{aligned}$$

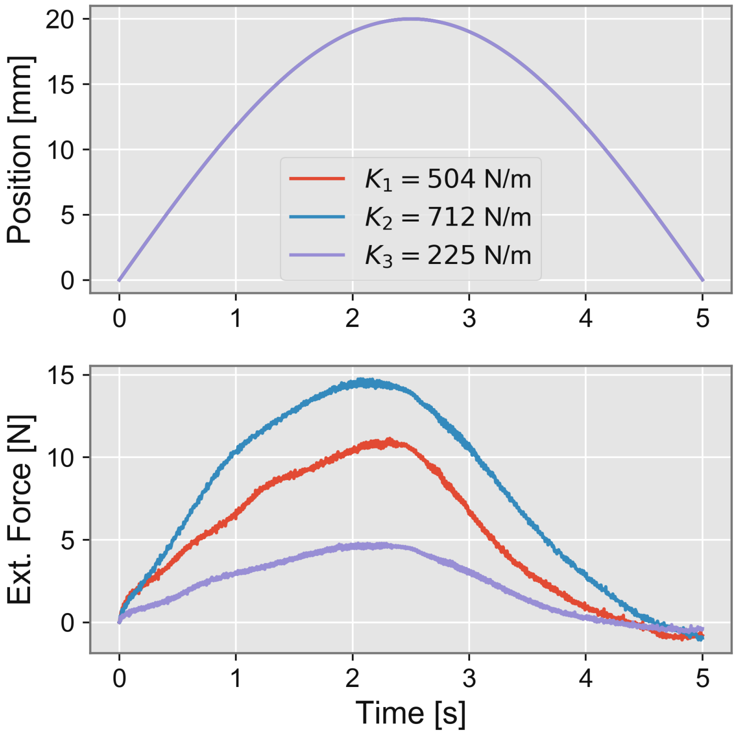


Figure 64: Target stiffness estimation – Position control

Force Control

Similar to what was done with the position control, the target estimation with force control is achieved by disabling the position control and using

$$F_m^* = -5 \sin(0.2\pi t)$$

as the bilateral master force reference. The negative sign is justified by fact that the bilateral controller objective is to obtain the overall force balance. As briefly described before, the force control alone is not stable: to avoid any damage to the experimental setup, at every control step the motor velocity is compared with a threshold value: if $v > 3\,\text{cm/s}$ the control command is zeroed and the exploration procedure interrupted. The results are reported in Figure 65. The maximum motor speed depends on the stiffness K:

$$F(t) = 5\sin(0.2\pi t)$$
$$x(t) = \frac{5\sin(0.2\pi t)}{K}$$
$$\max(\dot{x}) = \dot{x}(t=0) = \frac{\pi}{K}$$

with $K = 200\,\mathrm{N/m}$:

$$\max(\dot{x}) = \pi/200 = 1.57\,\mathrm{cm/s}$$

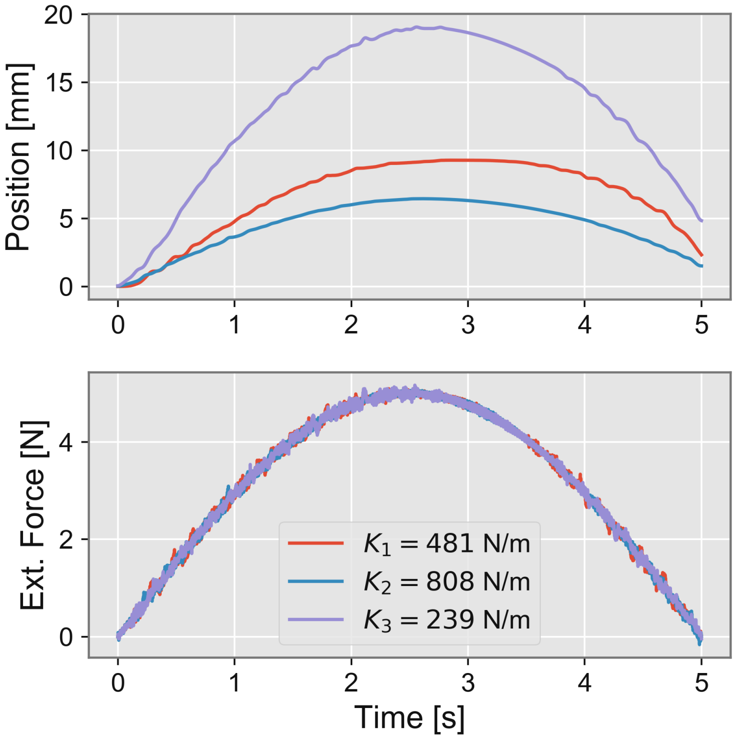


Figure 65: Target stiffness estimation – Force control

Comparison of the Stiffness Estimation Methods

To evaluate the two environment stiffness estimation procedures, the *coefficient of determination* R^2 is calculated for both. The coefficient of determination provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{105}$$

where SS_{res} is the sum of squares of the residuals

$$SS_{res} = \sum_i \left(F_i - \hat{K}\,\Delta x_i\right)^2$$

SS_{tot} the total sum of squares

$$SS_{tot} = \sum_i (F_i - \overline{F})^2$$

and $\overline{F}$ the mean force. The results are reported in Table 12 as well as the relative difference between the two methods.

Target	Posit. Ctrl [N/m]	Posit. Ctrl R^2	Force Ctrl [N/m]	Force Ctrl R^2	Difference
1	504	0.78	481	0.87	4.7 %
2	712	0.85	808	0.95	11.8 %
3	225	0.80	268	0.94	16.0 %

Table 12: Stiffness estimation comparison

While the environmental conditions didn't change between the two methods execution, from the calculated coefficients of determination the position/force data collected during the force control better match the proposed environment model in respect to the other method. The model expects the force to be proportional to the displacement, but in reality there is an hysteresis between the forward and backward movements: during the first half the motor pushes against the target, while during the second half there isn't any mechanical link between end-effector and target. The asymmetric behavior can be seen in Figure 64.

On the other hand, during the force control (second method) the motor position is adjusted to generate the reference reaction force, Figure 65. The target elastic non-linearity can be observer in the position trajectories: they don't return to the starting point even if no force is being applied and therefore present a skewed trajectory.

4.5.4 Target Stiffness Modification

After the exploration procedure described in the last subsection, the original task is adapted to modifications in the target material/composition. To cope with stiffness changes two line of action are considered:

- scaling the force reference to maintain the position unaltered
- scaling the position reference to maintain the applied force unaltered

Force Reference Scaling

The first option is to scale the force goal g^f of the second action (the one active during the interaction with the environment) proportionally with the target stiffness. Having defined the original target with A, the softer one with B and the stiffer one with C, the new force goals values are:

$$g^f_B = \frac{K_B}{K_A} g^f_A \qquad g^f_C = \frac{K_C}{K_A} g^f_A$$

The bilateral system references and outputs for the 3 different targets are reported in Figure 66. The target used to record the operator task (A) is used to obtain the task reproduction performance baseline: the maximum position error is $\max|e_p^A| = 1.12\,\mathrm{mm}$, while the maximum force error is $\max|e_f^A| = 1.1\,\mathrm{N}$. For the other two targets:

- for the stiffer object (B) the maximum position error is $\max|e_p^B| = 0.77\,\mathrm{mm}$ and the force error $\max|e_f^B| = 1.1\,\mathrm{N}$;
- for the softer object (C) the maximum position error is $\max|e_p^C| = 1.44\,\mathrm{mm}$ and the force error $\max|e_f^C| = 1.2\,\mathrm{N}$.

From the experimental position and force data samples, the force reproduction is effective (the absolute error between the reference and the estimated external forces is similar across all three trials) while the position tracking error become smaller as the stiffness of the object increases: this behavior can be explained because, as the target stiffness increases, the same force error produce a smaller displacement error. Therefore soft objects require a more precise position/force references – and are therefore less robust to position and force mismatches – because the same force error translates to a larger position error.

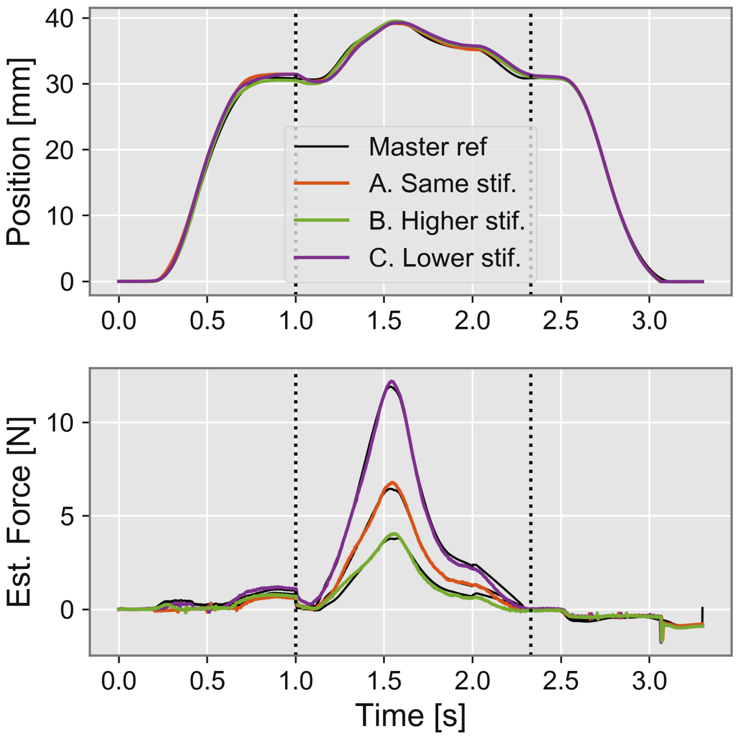


Figure 66: Environment adaptation – Force scaling to maintain the position unaltered.

Position Reference Scaling

The second option is to scale the position goal g^p of the second action, this time inversely proportional to the estimated target stiffness, to maintain the output force constant across different targets:

$$g_B^f = \frac{K_A}{K_B} g_A^p \qquad g_C^p = \frac{K_A}{K_C} g_A^p$$

The results are reported in Figure 67: as designed, the applied forces across the 3 different materials are close to the force reference, and, as observed in the previous method, the force error is similar over the three experiments: $\max|e_f^A| = 1.4\,\mathrm{N}$, $\max|e_f^B| = 1.4\,\mathrm{N}$, $\max|e_f^C| = 1.6\,\mathrm{N}$. The baseline performance, given by the original task and target, achieves a maximum absolute position error of $\max|e_p^A| = 0.34\,\mathrm{mm}$, while with the soft one the error is $\max|e_p^B| = 1.13\,\mathrm{mm}$ and the stiff one $\max|e_p^C| = 0.89\,\mathrm{mm}$. Similarly for what was shown with the previous method, while the force control achieves roughly the same performance independently of the target, the position error varies: the same force error translates to different position displacements, with higher stiffness materials generating smaller position errors.

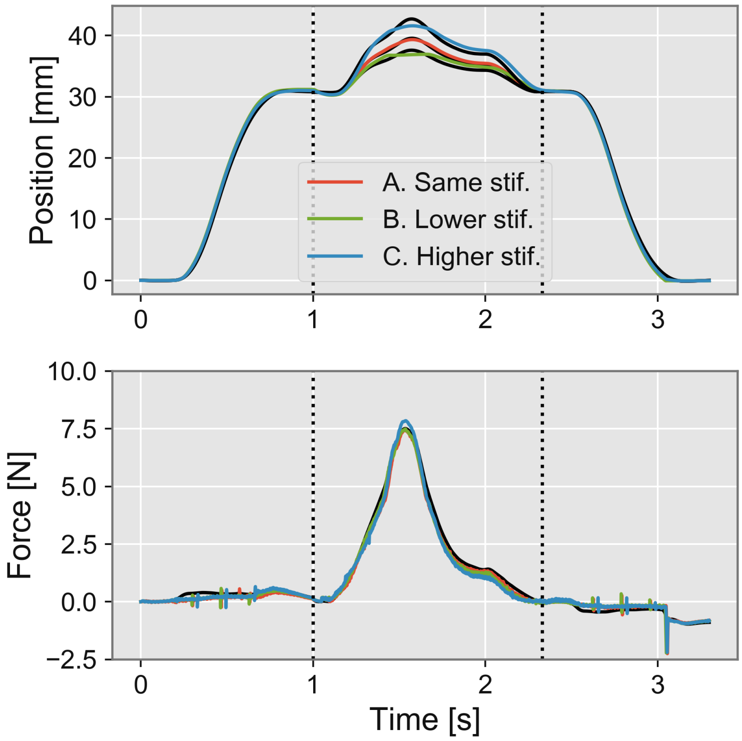


Figure 67: Environment adaptation – Position scaling to maintain a constant interaction force.

4.5.5 Final Comments

Several approaches to adapt a learned bilateral control action to new environmental conditions were proposed, as well as the mechanisms to estimate these changes. In particular:

- the action target position is detected by monitoring the estimated external force during a constant speed run – this method produces highly repeatable results and several parameters like the speed, force threshold and number of samples can be adjusted to obtain more precise results, faster estimation time or higher false-positives tolerance;

- the target stiffness is obtained by comparing the displacement and reaction force samples, that can be collected is several ways – the best results were obtained by controlling the applied force, ensuring that the is always contact between target and end-effector;
- the target position changes adaptation is performed by matching the DMP movement goal to the new target position;
- the task adaptation to a new target can be obtained by changing position or force references individually, depending on whether we need to maintain unaltered the applied force or the position path.

In the controlled experimental environment the task modification has always produces the desired effects by reducing the position/force errors, but the proposed approach should be extended and tested to multiple DOFs robots, operating over multiple degrees of freedom to further confirm the results.

4.6 ITERATIVE BILATERAL CONTROL

4.6.1 Iterative Learning Control

The Iterative Learning Control (ILC) paradigm was introduced for the control of systems that perform repetitive tasks and can therefore take advantage of the errors committed in the past iterations to improve the next executions performance. Defined T as the sampling period, $T_r = NT$ the repetitive task period and $k = 0, 1, 2, \dots, N-1$ each sample, the controlled system input generated by the ILC is:

$$u(n, k) = u(n-1, k) + K\, e(n, k) \tag{106}$$

in other words the control output generated in the previous cycle added to the current error multiplied by a proportional gain K. Over several task iterations, the control law $u(n, k),\ k = 0, 1, \dots, N-1$ is updated to incrementally reduce the system output error, starting from the null reference $u(0, k) = \mathbf{0}$ During the simulation and experimental trials, the control error decreases every iteration like predicted, but, reached a minimum after a number of cycles, starts to increase until the destabilization of the system. In fact the updating mechanism keeps adding to the control reference $u(n, k)$ the disturbances present in the feedback loop (e.g. sampling or simulation errors) multiplied by K, ultimately taking the system to instability. The solution is to limit the updating to a time window and then using the resulting $u(n, k)$ unaltered for the next $n + i$ cycles. When the learning needs to be resumed (for example because the external conditions have changed), it should start from the null reference $u(0, k)$ to avoid the error-adding complication.

4.6.2 Iterative Bilateral Control

In the previous sections the recorded bilateral control action were adapted to environment changes after an exploratory activity was performed to assess the new target position or stiffness. However, if the same action is performed multiple times over the same environment configuration, the position error committed in the previous runs can be used to improve the control reference, in a similar fashion to what the ILC suggests. The objective of the improvement is to minimize the position error while maintaining the force control, in other words to obtain a better position/force matching for the specific action. While the ILC records the control law as $u(n, k)$, in the bilateral control it is more convenient to directly modify the action virtual master force reference F_m, directly involved in the force controller. The proposed corrective action is presented below:

1. after each action execution, the recorded position and force samples are used to estimate the target stiffness $\hat{K}$ with the same model used in the subsection 4.5.3;

2. the force reference of the next execution $n + 1$ is supplemented with the position error multiplied by the estimated target stiffness;

$$F_m(n+1, k) = F_m(n, k) + \hat{K}(n) \cdot e_p(n, k)$$

3. the weights $\mathbf{w}^f$ of the action DMP model are updated with the new reference.

The same instability problems found in the ILC need to taken care of: to detect the minimum error configuration, the total quadratic error L_2 is calculated for each step and monitored:

$$L_2(n) = \sum_k (x_m(k) - x_s(k))^2 \tag{107}$$

when the error trend is decreasing the update process is active and both the current ($F_m(n,k)$) and the previous ($F_m(n-1,k)$) master force references are saved. As soon as the error starts increasing

$$L_2(n-1) < L_2(n)$$

the F_m updating is stopped and $F_m(n-1,k)$ is used as the master force reference.

Experimental Testing

The saved action used for the testing of the DMP scaling properties (subsection 4.4.1) is utilized, but the original target is substituted with a stiffer one, to introduce the (unknown) environmental changes. Each step main parameters (position error, loss function, target estimated stiffness) are reported in Table 13, position and force results in Figure 68.

The force weight values evolution across the different runs is presented in Figure 69, where we see a non-linear relationship between the force amplitude and the RBFs, justifying the fact that the force update is done before in the time domain and then the weights are updated to reflect the new force reference.

Step	Max. position error [mm]	Loss func. L_2	Est. Stiffness [N/m]
0	2.37	1.837	733
1	1.01	0.750	747
2	0.525	0.351	743
3	0.305	0.213	741
4	0.196	0.120	747
5	0.217	0.175	752

Table 13: Iterative bilateral control step results.

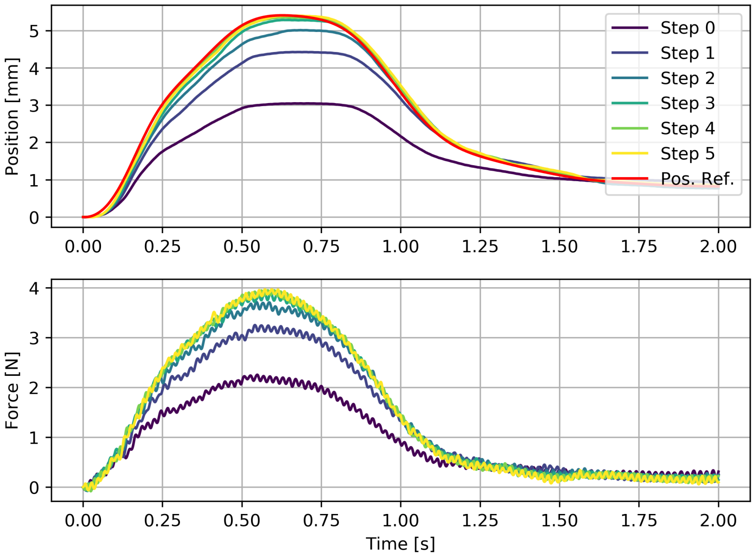


Figure 68: Position and force data from consecutive steps with the force reference updating.

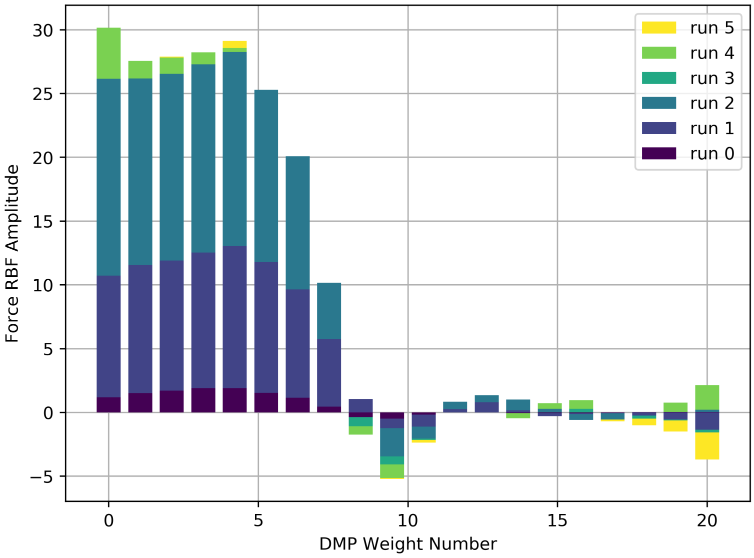


Figure 69: Force weight updating during the iterative learning.

CONCLUSIONS

The thesis starting point was the study of techniques to save the manual expertise of a person by recording a task motion position and applied forces. This system could be useful to store the person manual knowledge with the goal of reproducing said motion whenever necessary, without the constraint of the person presence, or to train future workers that weren't originally available.

To accurately record a task motion it is necessary to record both the movement path and the applied forces. This goal was accomplished in the second chapter by first studying and designing a Bilateral Control system, capable of reproducing the operator movement (master side) at the slave side, and then extending the system to record the motion and reproduce it using the recorded user position and forces as a virtual master reference, realizing a Motion Copying System. In the second chapter the MCS stability was assessed by studying the transfer function poles between the saved and the reproduced force: the system is stable even with huge variations of the environment conditions. As easily predicted the presence of environmental changes leads to reproduced position and force errors, that degrade the system ability to accomplish the goal of accurate motion reproduction. In the third chapter a solution to the problem of bilateral task environment adaptation is proposed; the considered variations are limited to changes in the target distance and/or its stiffness. Another problem with the MCS references is the large number of samples recorded and stored to represent a task. A model (based on the Dynamic Movement Primitives) was introduced to simplify the scaling of a saved task and to reduce the amount of data used to represent an action, with the cost of an abstraction layer introduced by the model dynamic system and forcing terms.

Several ways of adapting the saved task to the environment have been proposed to estimate the target position and stiffness. The new information about the environment is successfully used to (linearly) scale position/speed and force reference to adapt the task and reduce the mismatch error. The environment model used is simple: it assumes that the environmental damping is negligible and the stiffness is constant.

Another option to adapt the task to different conditions, briefly explored in the last section, is based on the Iterative Learning Control. It assumes that the considered task needs to be reproduced several times and, at every execution, corrects the force reference to reduce the position error by estimating the environment stiffness and supplementing the force reference with the estimated difference.

While the proposed approaches were successfully implemented and always led to the reduction of position and force errors, validating the approach and the implementation, the experimental conditions were as simple as they could get: one degree of freedom linear motor and environment model with negligible damping coefficients.

Unfortunately during the developing of this thesis I wasn't able to extend the proposed approach to multiply degrees of freedom manipulators, so this should be the next development step. In particular, by using a cartesian robot, the DOF dynamics result for the most part decoupled. Increasing the linear motor travel should be easily achievable by modifying the experimental setup. Another development I would like to study is the broadening of environmental conditions under which the adapting mechanism is tested, in particular extended the proposed approaches to include a damping effect or in the case of non-linear environment mechanical properties.

A BILATERAL CONTROL REFERENCE SCALING

In the first chapter (section 2.3) the bilateral control reference scaling was briefly introduced. This mechanism consist of scaling the position/speed or force signals of a single bilateral controller channel (master or slave) to achieve a virtual force or movement amplification. This method can be used to lighten the operator applied force or increase the movement precision.

The bilateral controller inputs are the master and slave differential position/speed $x_d, \dot{x}_d$ and the common estimated force F_c. Two scaling coefficients are introduced: α on the slave side position/speed and β on the estimated force.

$$\begin{bmatrix} x_c \\ x_d \end{bmatrix} = Q_2 \begin{bmatrix} 1 \cdot x_m \\ \alpha \cdot x_s \end{bmatrix}$$

$$\begin{bmatrix} F_c \\ F_d \end{bmatrix} = Q_2 \begin{bmatrix} 1 \cdot F_m \\ \beta \cdot F_s \end{bmatrix}$$

Because the bilateral controller is symmetrical from the point of view of which system is the master and which is the slave, the scaling is introduced only on one side of the system.

A.1 POSITION SCALING

The value of α represent the ratio between the master and slave movement amplitude and speed.

$$x_m = \alpha\, x_s$$

An experimental validation is done for $\alpha = 0.5$ in Figure 70 and $\alpha = 2$ in Figure 71. As expected, from the collected samples during the bilateral control the master amplitude is half of the slave's for $\alpha = 0.5$ and doubled for $\alpha = 2$, but the applied forces balance remains unvaried.

The obtained system is very different from what a normal gearbox would produce: only the position and speed components are affected, but not the force, while the ideal gearbox (for the input/output power balance) would scale the position and transmitted forces inversely proportional. This also means that the reproduced impedance doesn't match the environment at the slave side:

$$Z_m = \frac{F_m}{x_m} = \frac{F_s}{\alpha\, x_s} = \frac{Z_s}{\alpha}$$

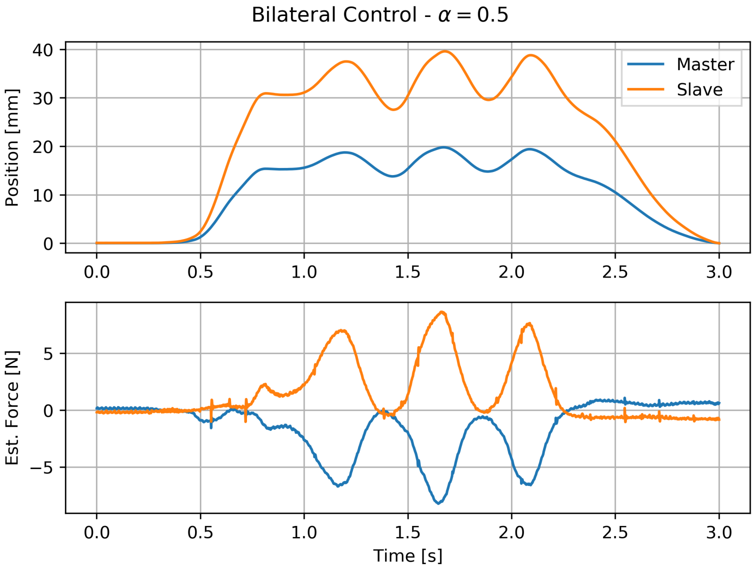


Figure 70: Bilateral control under the position and speed scaling with $\alpha = 0.5$ – the slave movement amplitude is doubled in respect to the master's.

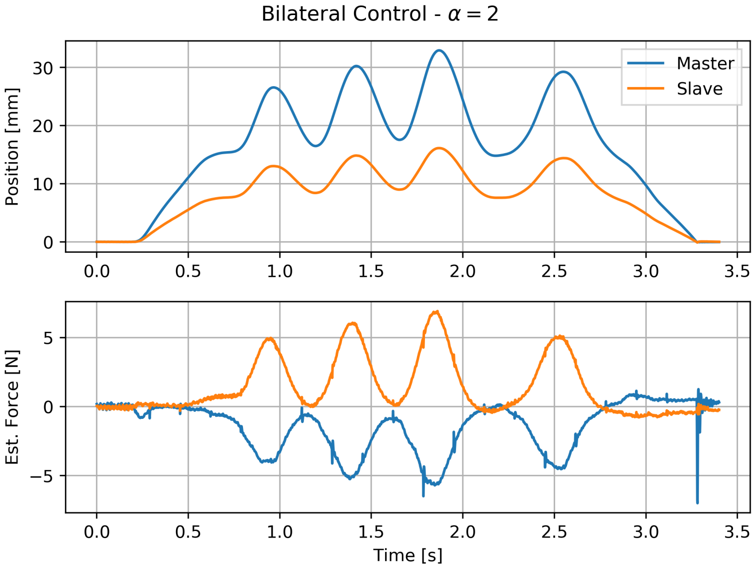


Figure 71: Bilateral control under the position and speed scaling with $\alpha = 2$ – the slave movement amplitude is halved in respect to the master's.

A.2 FORCE SCALING

Through the β coefficient is possible to adjust the master/slave force balance ratio, like was done for the position scaling. Specifically:

$$F_m = \beta\ F_s$$

In Figure 72 is presented the experimental data with $\beta = 0.5$. Like for the position scaling, also in this configuration the reproduced impedance at the master side doesn't match slave side one:

$$Z_m = \frac{F_m}{x_m} = \frac{\beta\ F_s}{x_s} = \beta\ Z_s$$

The two scaling factor can be chosen jointly to realize a specific objective:

- with $\alpha = \beta$ we obtain the perfect environment impedance reproduction because the movement amplitude and applied forces are scaled proportionally
$$Z_m = \frac{F_m}{X_m} = \frac{\beta\ F_s}{\alpha\ X_s} = Z_s$$
- with $\alpha = 1/\beta$ the system mimics the action of a gear reduction mechanism
$$Z_m = \frac{F_m}{X_m} = \frac{\beta\ F_s}{X_s/\beta} = \alpha^2\ Z_s$$

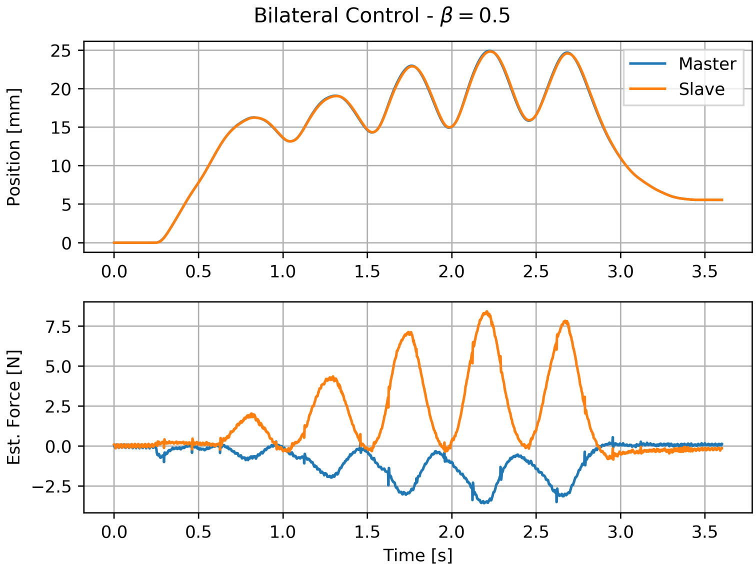


Figure 72: Bilateral control under the force scaling with $\beta = 0.5$ – the slave force amplitude is double the master's.

BIBLIOGRAPHY

[1] Sameer M. Prabhu and Devendra P. Garg. "Artificial neural network based robot control: An overview". In: *Journal of Intelligent and Robotic Systems* 15.4 (Apr. 1996), pp. 333–365.

[2] Long Jin et al. "Robot manipulator control using neural networks: A survey". In: *Neurocomputing* 285 (2018), pp. 23–34.

[3] Stefano Ghidoni. "Robotic Arm Control and Task Training through Deep Reinforcement Learning". In: *journaltitle* ().

[4] Stefan Schaal et al. "Control, Planning, Learning, and Imitation with Dynamic Movement Primitives". In: *Workshop on Bilateral Paradigms on Humans and Humanoids, 2003 IEEE International Conference on Intelligent Robots and Systems IROS* (2003), pp. 1–21.

[5] Yuki Yokokura, Seiichiro Katsura, and Kiyoshi Ohishi. "Stability analysis and experimental validation of a motion-copying system". In: *IEEE Transactions on Industrial Electronics* 56.10 (2009), pp. 3906–3913.

[6] Kenji Natori, Roberto Oboe, and Kouhei Ohnishi. "Stability analysis and practical design procedure of time delayed control systems with communication disturbance observer". In: *IEEE Transactions on Industrial Informatics* 4.3 (2008), pp. 185–197.

[7] Noboru Tsunashima and Seiichiro Katsura. "Spatiotemporal coupler: Storage and reproduction of human finger motions". In: *IEEE Transactions on Industrial Electronics* 59.2 (2012), pp. 1074–1085.

[8] W. Iida and K. Ohnishi. "Reproducibility and Operationality in Bilateral Teleoperation". In: *The 8th IEEE International Workshop on Advanced Motion Control* (2004).

[9] Dale A. Lawrence. "Stability and Transparency in Bilateral Teleoperation". In: *IEEE Transactions on Robotics and Automation* 9.5 (1993), pp. 624–637.

[10] K. Hastrudi-Zaad and S.E. Salcudean. "On the use of local force feedback for transparent teleoperation". In: 2003.

[11] Blake Hannaford. "A Design Framework for Teleoperators with Kinesthetic Feedback". In: *IEEE Transactions on Robotics and Automation* (1989).

[12] K. M. DeGoede et al. "How quickly can healthy adults move their hands to intercept an approaching object? Age and gender effects". In: *Journals of Gerontology - Series A Biological Sciences and Medical Sciences* 56.9 (2001), pp. 584–588.

[13] Yoshiyuki Kambara, Seiji Uozumi, and Kouhei Ohnishi. "Stability analysis of bilateral control by considering environmental impedance and modeling error". In: *Proceedings of the IEEE International Conference on Industrial Technology* 2015-June.June (2015), pp. 481–486.

[14] Roberto Oboe. "How disturbance observer changed my life". In: *Proceedings - 2018 IEEE 15th International Workshop on Advanced Motion Control, AMC 2018* (2018), pp. 13–20.

[15] Stefan Schaal. "Dynamic Movement Primitives -A Framework for Motor Control in Humans and Humanoid Robotics". In: *Adaptive Motion of Animals and Machines* (2002), pp. 261–280.

[16] Auke Jan Ijspeert et al. "Dynamical movement primitives: Learning attractor models formotor behaviors". In: *Neural Computation* 25.2 (2013), pp. 328–373.

[17] Jens Kober and Jan Peters. "Learning motor primitives for robotics". In: *2009 IEEE International Conference on Robotics and Automation* (2009), pp. 2112–2118.

[18] Ian D. Loram, Peter J. Gawthrop, and Martin Lakie. "The frequency of human, manual adjustments in balancing an inverted pendulum is constrained by intrinsic physiological factors". In: *Journal of Physiology* 577.1 (2006), pp. 417–432.

[19] H. J. Freund. "Time control of hand movements". In: *Progress in Brain Research* 64.C (1986), pp. 287–294.

[20] Shunsuke Yajima and Seiichiro Katsura. "Multi-DOF motion reproduction using motion-copying system with velocity constraint". In: *IEEE Transactions on Industrial Electronics* 61.7 (2014), pp. 3765–3775.

[21] N Tsunashima and S Katsura. "Continuous integration of motion components using motion copying system". In: *Iccas-Sice* (2009), pp. 1569–1574.

[22] Nutan Chen, Maximilian Karl, and Patrick Van Der Smagt. "Dynamic movement primitives in latent space of time-dependent variational autoencoders". In: *IEEE-RAS International Conference on Humanoid Robots* (2013), pp. 629–636.

[23] Mark J.L. Mjl Orr. "Introduction to radial basis function networks". In: *University of Edinburg* (1996), pp. 1–7.

[24] Seiichiro Katsura and Kiyoshi Ohishi. "Modal system design of multirobot systems by interaction mode control". In: *IEEE Transactions on Industrial Electronics* 54.3 (2007), pp. 1537–1546.

[25] Nariman Roshandel Tavana et al. "Design of Slotted Permanent Magnet Linear Synchronous Motor for Improved Thrust Density". In: (2013), pp. 1225–1228.

[26] Yuki Yokokura, Seiichiro Katsura, and Kiyoshi Ohishi. "Motion copying system based on real-world haptics in variable speed". In: *2008 13th International Power Electronics and Motion Control Conference, EPE-PEMC 2008* (2008), pp. 1604–1609.

[27] Kazumasa Miura, Ayaka Matsui, and Seiichiro Katsura. "High-Stiff Motion Reproduction Using Position-Based Motion-Copying System with Acceleration-Based Bilateral Control". In: *IEEE Transactions on Industrial Electronics* 62.12 (2015), pp. 7631–7642.

[28] Shunsuke Yajima and Seiichiro Katsura. “Towards performance improvement of motion reproduction based on motion-copying system”. In: *2013 IEEE International Conference on Mechatronics, ICM 2013* (2013), pp. 582–587.

[29] He Zhang et al. “Research on the End Effect Detent Force Reduction of Permanent Magnet Linear Synchronous Motor with Auxiliary Poles One-Piece Structure”. In: *Applied Mechanics and Materials* 416-417 (2013), pp. 27–32.

[30] Seiichiro Katsura, Yuichi Matsumoto, and Kouhei Ohnishi. “Realization of "law of action and reaction" by multilateral control”. In: *IEEE Transactions on Industrial Electronics* 52.5 (2005), pp. 1196–1205.

[31] Seiichiro Katsura, Wataru Yamanouchi, and Yuki Yokokura. “Real-world haptics: Reproduction of human motion”. In: *IEEE Industrial Electronics Magazine* 6.1 (2012), pp. 25–31.

[32] Shi Uk Chung et al. “Force ripple and magnetic unbalance reduction design for doubly salient permanent magnet linear synchronous motor”. In: *IEEE Transactions on Magnetics* 47.10 (2011), pp. 4207–4210.

[33] A. Paraschos et al. “Using probabilistic movement primitives in robotics”. In: *Advances in Neural Information Processing Systems* (2013), pp. 2616–2624.

[34] Yuki Nagatsu and Seiichiro Katsura. “Decoupling and Performance Enhancement of Hybrid Control for Motion-Copying System”. In: *IEEE Transactions on Industrial Electronics* 64.1 (2017), pp. 420–431.

[35] Jens Kober, J Andrew Bagnell, and Jan Peters. “Reinforcement learning in robotics: A survey”. In: *The International Journal of Robotics Research* 32.11 (Sept. 2013), pp. 1238–1274.

[36] Jens Kober, Betty Mohler, and Jan Peters. “Learning perceptual coupling for motor primitives”. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS* (2008), pp. 834–839.

[37] Jens Kober and Jan Peters. “Policy Search for Motor Primitives in Robotics”. In: *Advances in Neural Information Processing Systems (NIPS)* 84.1-2 (2011), pp. 171–203.

[38] Freek Stulp et al. “Model-free reinforcement learning of impedance control in stochastic environments”. In: *IEEE Transactions on Autonomous Mental Development* 4.4 (2012), pp. 330–341.

[39] Stefan Schaal, Christopher G. Atkeson, and Sethu Vijayakumar. “Scalable techniques from nonparametric statistics for real time robot learning”. In: *Applied Intelligence* 17.1 (2002), pp. 49–60.

[40] Kazumasa Miura, Ayaka Matsui, and Seiichiro Katsura. “Synthesis of motion-reproduction systems based on motion-copying system considering control stiffness”. In: *IEEE/ASME Transactions on Mechatronics* 21.2 (2016), pp. 1015–1023.

[41] Yuki Yokokura and Seiichiro Katsura. “Adaptive motion-copying system based on real-world haptics”. In: *IECON Proceedings (Industrial Electronics Conference)*. 2010, pp. 1228–1233.

[42] Yuki Nagatsu and Seiichiro Katsura. "Design strategies for motion reproduction based on environmental disturbance compensation". In: *IEEE Transactions on Industrial Electronics* 62.9 (2015), pp. 5786–5798.

[43] Yuki Yokokura, Kiyoshi Ohishi, and Seiichiro Katsura. "Fine force reproduction based on motion-copying system using acceleration observer". In: *IEEE Transactions on Industrial Electronics* 61.11 (2014), pp. 6213–6221.

[44] Kazuki Nagase et al. "Advanced motion copying system of multi degree-of-freedom human motion". In: *Proceedings - The 12th International Conference on Electrical Machines and Systems, ICEMS 2009* (2009), pp. 1–6.