POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in
Ingegneria Meccanica

Sensorless Kinesthetic Teaching of Robotic Manipulators Assisted by an Observer-based Force Control

Relatore: Prof. Paolo Rocco

Correlatori: Prof. Rolf Johansson
Prof. Anders Robertsson
Dr. M. Mahdi Ghazaei

Tesi di Laurea di:
Martino Capurso
Matr. 819807

Anno Accademico 2015 - 2016

Acknowledgments

Many thanks to my supervisor Prof. Paolo Rocco, who gave me the opportunity to develop this challenging thesis project in a European context.

I am grateful to my Swedish supervisors Prof. Rolf Johansson and Prof. Anders Robertsson, who welcomed me in the Automatic Control Department of Lund University and advised me during my research, as well as to Dr. Eva Westin for her efficient and friendly support.

Special thanks go to my advisor Dr. M. Mahdi Ghazaei, who guided and supported me with patience during my stay in Lund.

I cannot find the words to thank my family, who has always been present and motivating, supporting me through all my goals and achievements. Thank you so much.

Abstract

In the modern industry, the use of industrial robots is fundamental to achieve high production rates and competitiveness. In order to employ automatic processes also when the product changes rapidly, e.g. as in short-series production, it is vital to be able to reprogram robots quickly. A fast programming method is the kinesthetic teaching, also called lead-through programming (LTP), that teaches the robot a trajectory by physical demonstration, i.e. the user is able to manually guide the manipulator during the programming phase. This thesis presents an active LTP implemented with a sensorless approach, so that the usually expensive force/torque sensors are not needed. The active control of the manipulator is achieved via an admittance control implemented in joint space, based on the external forces applied by the user. The external torques are estimated with a Kalman filter, using the generalized momentum formulation. The estimate of the external torques is model-based, thus, a dynamic model of the robot is presented, with particular attention to joint friction modeling. The final LTP algorithm also includes a model-based feedforward compensation of gravity load and joint friction. The active LTP has been implemented on the robot ABB YuMi; experimental tests regarding force estimation and the active control performances are shown. Furthermore, an experimental comparison with a passive LTP is presented.

Key words: robotics; teaching by demonstration; friction models; disturbance observer

Sommario

Nei moderni impianti industriali, un elevato grado di automazione è indispensabile al fine di garantire elevata produttività e competitività. Per incrementare l'utilizzo dei robot industriali in impianti destinati a piccole produzioni, i quali devono essere facilmente modificabili per adeguarsi ai cambiamenti di prodotto, la fase di programmazione del robot deve risultare il più breve possibile. Il *kinesthetic teaching*, anche detto *lead-through programming* (LTP), rappresenta un metodo di programmazione rapida basato sull'insegnamento tramite dimostrazione, in cui l'operatore può muovere fisicamente il manipolatore per insegnare la traiettoria che verrà successivamente ripetuta in modo autonomo. In questa tesi è presentato un LTP attivo implementato senza l'uso di sensori, i quali solitamente risultano costosi e fragili. Il controllo di forza in retroazione è implementato tramite un controllo in ammettenza nello spazio dei giunti, sulla base delle coppie esterne stimate con un filtro di Kalman che utilizza il momento generalizzato. La stima delle forze esterne applicate dall'operatore si basa sul modello del robot, viene quindi presentato un modello dinamico del manipolatore con particolare attenzione alla modellazione dell'attrito nei giunti. Per la parte sperimentale della tesi è stato utilizzato il robot ABB YuMi; sono successivamente presentate le prove sperimentali riguardanti la compensazione di attrito e forza gravitazionale, la stima delle coppie esterne e il controllo in ammettenza. Infine, viene proposto un confronto tra l'LTP attivo sviluppato in questa tesi e una versione passiva implementata in precedenza.

Parole chiave: robotica; programmazione per apprendimento; modelli di attrito; osservatore del disturbo

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the modern manufacturing industry, the employment of industrial robots is constantly increasing, and the role of modern technologies in industrial processes is vital to achieve high competitiveness. Industrial robots, in particular robotic manipulators, are replacing human workers especially in those tasks with low complexity and high repeatability, to increase productivity and accuracy.

To introduce robotic manipulators also in short-series productions, in which the production line has to be often reconfigured, new methodologies to reduce the initial programming time of the robot, and therefore to reduce operative costs, have to be developed. An intuitive way to accelerate the programming phase of the robot, and therefore reduce production down-time, is to manually guide the robot, which is usually called lead-through programming (LTP) or kinesthetic teaching. These techniques aim to teach the trajectory by manual handling of the manipulator, so that the robot will repeat it autonomously in a later time.

Many methodologies and techniques to realize the LTP have been developed; a raw characterization distinguishes the passive and the active LTP. The former approach is usually performed by feedforward compensation of gravity and friction. The latter includes a force-feedback loop based on the knowledge of the external torques applied by the user. In order to detect the external torques, a common solution is to install torque/force sensors on the robot, sensors which are generally very expensive and fragile; moreover, they add mass to the robot reducing the maximum payload. A cheaper method to realize an active LTP is based on the sensorless estimation of external forces. Sensorless methodologies usually make use of control algorithms (force observers) based on the dynamic model of the manipulator, implemented in Cartesian or joint space.

In this thesis, the anthropomorphic manipulator that was used for

the experimental part is the ABB YuMi robot, shown in Figure 1.1, a small dual arm industrial robot with 7 degrees of freedom in each arm. This robot is designed to operate in an industrial environment shared with human workers; in fact, it has lightweight links covered with soft paddings, in order to guarantee a safe human-machine interaction.

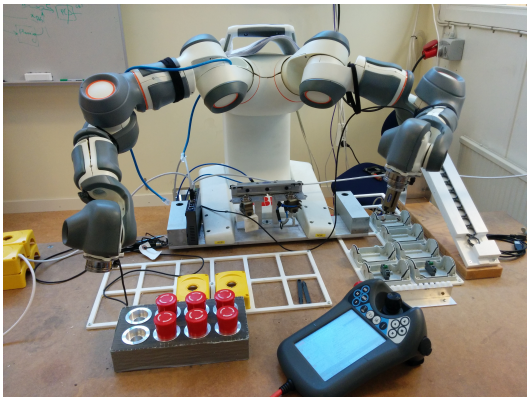


Figure 1.1: The ABB YuMi robot in the Robotics Lab of Lund University used for the experiments.

1.1 Problem Formulation

The final goal of this thesis is to develop a sensorless active lead-through programming interface, that aims to assist the operator in the handling of the manipulator thanks to a force-feedback control. The active control can have multiple purposes, from the simple support during the movement achieved by a motor torque that follows the operator's intentions, to the implementation of virtual constraints to limit the work-space or to accomplish obstacles avoidance.

In order to implement the active lead-through programming, two main sub-problems can be distinguished: the force control, that determines the robot response to the external forces, and the force observer, for external torque estimation without using sensors. The force control can be accomplished by an indirect admittance control, in which the external torques on the joints represent the input and the resulting desired displacements and velocities of the robot joints are the output.

As concern the external force detection, various sensorless schemes have been suggested relying on force estimation techniques. In general, there are two main approaches for estimating external forces: using a disturbance observer based on the control error, or a force observer utilizing

in addition the knowledge of motor torques. A convenient approach is based on the generalized momentum formulation, since the equation of motion of the robot can be written in a state-space system where neither the inversion of the inertia matrix nor the calculation of the numerical acceleration are required.

The implementation in joint space is preferable, i.e. each joint of the robot is controlled independently, so that no problems with singularities of the robot will occur. This is particularly advisable in redundant manipulators like YuMi.

In order to design a stable and smooth controller, it is convenient to implement a feedforward compensation of gravity load and joint friction. A dynamic model of YuMi is then necessary to calculate the gravity torques, in particular an accurate friction model will be fundamental also for the implementation of the disturbance observer.

1.2 Methodology

The first phase of this thesis concerns the definition of a dynamic model of the robot ABB YuMi, that will be achieved with the support of Peter Corke's robotic toolbox [11]. A dynamic model is necessary for simulations and to calculate the gravity compensation and, overall, it is crucial for the implementation of the force observer. In particular, the kinematics can be defined using the Denavit-Hartenberg notation, given the lengths of links; the dynamic parameter of each link (center of mass, mass and inertia) are provided with a data sheet. The joint friction is unknown, therefore, a campaign of experiments will be performed in order to define a reliable friction model. The successively experimental validation of the model is a significant phase of the project; for this purpose, experimental and simulated data for a given trajectory will be compared, to verify the matching in term of position, torques and velocities.

Before designing the active force controller, a passive LTP via feedforward compensation will be firstly implemented. The low-level control loops of the native ABB controller will be disabled setting the control gains equal to zero, so that the robot can be manually moved while compensating for gravity and friction with a feedforward motor torque.

Once a reliable dynamic model of YuMi is available, it is possible to design a disturbance observer for force estimation. A strategy based on a Kalman-Bucy filter implemented with the generalized momentum approach is chosen. In fact, the generalized momentum formulation does not required neither the inversion of the inertia matrix nor the calculation

of the numerical acceleration. To estimate the external torques, the measurements of joint position and velocity, together with the motor torques, are required. After the simulation phase, an experimental tuning of the observer parameters will be necessary.

The active LTP will be implemented via admittance control, based on the estimation of the external torques. The behavior of the robot will then be controlled in terms of damping and inertia, while compensating for the gravity force with a feedforward component. An iterative phase between simulations and experiments, in order to tune the controller and to test the performance, is expected.

In the final phase of the project, an experimental comparison between the active LTP illustrated in this thesis and a passive version already available in the Robotics Lab will be performed, in order to test the improvements given by the active implementation.

1.3 Chapters Organization

The next chapters are organized as follow:

- **Chapter 2** summarizes the state of art of the most recent programming method algorithms via human-robot interaction. Particular attention is payed on sensorless lead-through programming algorithms that employ force estimation approaches.

- **Chapter 3** concerns the definition of the dynamic model of the ABB YuMi robot. The first section will describe the experimental set-up that was used for the experiments.

- **Chapter 4** illustrates a passive LTP implemented via feedforward gravity and friction compensation. To accomplish the compensation of part of the joint friction, an accurate friction model is essential, and its definition is presented in the first section.

- **Chapter 5** describes the force observer that has been implemented to estimate the external torques applied by the operator. A theoretical dissertation firstly explains the algorithm, while successively the implementation and the experimental tests are presented. The performance and the major limitations are discussed.

- **Chapter 6** presents the admittance control architecture that has been implemented to make the LTP active. Several experiments

point out the controller performance, and finally a quantitative comparison with a passive LTP is shown.

- **Chapter 7** concerns the conclusions of this thesis project. Possible future developments are proposed.

Chapter 2

State of Art

The modern industrial production is characterized by increasing dynamics of innovation, shortened product life cycles, and a continuing diversification of the product range. Considering the high cost of skilled workers, the industrial automation based on robotic systems represents the best solution for both productivity and flexibility. A remarkable limitation for the application of automated systems is the time required for the programming phase, that makes this option practically unrealizable for small and medium production volumes.

As concern robot programming methodologies for industrial applications, a lot of work have been done in the last twenty years. The next sections present a summary that aims to characterize the majority of the robot programming methods and to illustrate the most investigated strategies of lead-through programming.

2.1 Programming Methods Overview

In this section, the main techniques for robot programming are briefly categorized.

An overview of different programming methods for industrial robots is presented in [20]. According to the article, two main categories of robot programming approaches can be distinguished: on-line programming (including lead-through and walk-through) and off-line programming (OLP).

Generally, for on-line programming the teach pendant or demonstration are used to manually move the end-effector along the desired trajectory; at the same time, the joints positions are recorded and processed by the robot controller in order to define the work parameters. The on-line programming is usually suitable for easy tasks onto work-pieces with

simple geometry, it is fast and intuitive and does not require high programming skills.

The OLP methods are usually based on the 3D model of the complete robot work cell. The OLP is generally preferable for complex programming tasks and it is particularly convenient for production with large volumes. Compared to the on-line programming method, it is more reliable and provides certain flexibility to the changes of product design, on the other hand, it is time consuming and requires skilled operators.

A modern tendency in the research environment is to combine the OLP and on-line approaches, i.e. to mix the knowledge of real world and the CAD model together. An example can be found in the robot programming using augmented reality (RPAR), that is a recently developed approach that aims to improve the intuitiveness and flexibility of OLP task. These techniques combine human-robot interaction, sensor technology and CAD systems, consequently, the difference between on-line and off-line approaches is not well defined anymore.

2.2 Programming by Demonstration

On-line programming was conventionally carried out by skilled operators, that were able to define the desired trajectory moving the manipulator using the teach pendant. Typically, the jogging of the robot is a delicate task, especially when the work piece has a complex geometry or the process itself is very complicated. Moreover, sometimes it might not be intuitive, since in a robotic system many coordinate frames are usually defined. In addition, some drawbacks of the method include that the operator is exposed to a hostile environment, the robot cannot be used for production during the teaching period, and the quality of the programming relies on the operator's ability. Nevertheless, the lead-through programming is a largely diffused method, in particular for short-series production, in which the programming phase has to be very fast.

According to Pan's article [20], the assisted on-line programming is categorized into "operator assisted on-line programming" and "sensor guided on-line programming". The former approach is based on teaching support devices integrated on 3D graphical user interfaces. The latter usually involves hybrid position/force/vision control platforms, developed to control the robot motion in different directions using various sensor feedbacks. Stereo vision is also used to acquire 3D coordinates for robot programming; for instance, it can be used to easily identify corners and edges from a work piece.

In literature there are few studies, mostly experimental studies, for investigating practical stability problems in human-robot cooperative operations. Stability issues can arise in several scenarios, for instance, because of the contact with a high stiffness environment or due to the time delay between the operator reaction and the robot controller in the cooperative task. In [27] is investigated the stability of the impedance control of a specific robot system in relation with human interaction. A 1 DOF model, shown in Figure 2.1, plus the compliance at the end-effector are considered. Although the human-robot interaction is very hard to model properly, the experimental results illustrate that the investigated stability analysis is effective to simulate and reveal the stability of the human-robot cooperative task system.

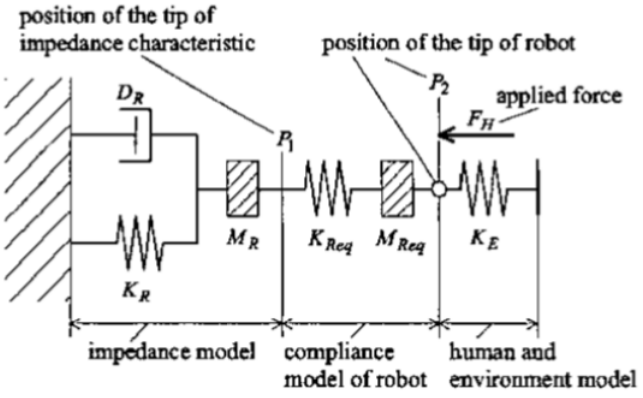


Figure 2.1: Simplified model of the robot used in [27] for stability studies concerning human-robot interaction.

An effective application of LTP is presented in [6], where LTP was used to simplify the teaching of weld paths in a shipyard. To accomplish the LTP, the algorithm was based on impedance control with zero stiffness, to allow the robot to be moved by hands during walk-through teaching. The robot was equipped with a force/torque sensor. Test for performance evaluation confirmed that the quality of the weld accomplished by the robotic system was better and faster than the manual, while the programming phase was shorter than programming by teach pendant.

Another interesting implementation of an active LTP is presented in [10], in which "the robot first observes the task performed by the user (through motion sensors) and the robot's skill is then refined progressively by embodying the robot and putting it through the motion (kinesthetic teaching)". In this example the human teacher is included in the

loop of the robot's learning, resulting in an incremental learning approach that permits to interact and improve the robot's movement, iterating the demonstration phase. The article proves the effectiveness of demonstration through motion sensors (contact-less) and the statistical enhancement of the performance based on multiple demonstrations.

In [30] attention is focused on redundant robotic manipulators, where a system concept that includes the possibility of teaching also the geometrical constraints of the task is studied. In this case, a two-step process is implemented: firstly the task constraints are taught by demonstration, secondly the user teaches the desired trajectory (by physical demonstration) with the support of a hierarchical control that takes into account the learned constraints. In the experiments, the authors demonstrated that the configuration of the redundancy resolution through kinesthetic teaching and the learning of the inverse kinematics mapping can be done in less than 2 minutes. Furthermore, the separation of two-phases teaching (firstly the constraints teaching, secondly the task teaching) permits to save time and to obtain a reconfigurable redundant robot.

In [7] the authors presented a walk-through programming implemented with an admittance control, based on the measures of a force sensor end-effector-mounted. An accurate tool dynamic model is illustrated, in order to reproduce the impression of physical interaction with the tool during the LTP. The safety aspects are here underlined, in fact, two strategies to constrain the robot motion are presented: the former limits the robot Cartesian velocity, the latter establishes a virtual space constraint to limit the work-space of the robot so to protect the operator.

An example of the potentiality of the combination of disturbance sensing and force control is illustrated in [15]. In this article, a light weight robot made in the German Aerospace Center (DLR) has been equipped with force sensors on each joint (Figure 2.2). Several scenarios of human-robot interaction are presented, e.g. teaching by demonstration, obstacles avoidance and unforeseen contacts. An interesting case is the self motion of the redundant DLR robot arm influenced by user interaction: when an external torque on a joint is detected, the manipulator reacts by an evasiveness null-space motion.

2.3 Sensorless LTP

Many recent studies have been carried out on sensorless programming approaches, in order to eliminate the need for sensors and, therefore, reduce costs. Sensorless implementations are usually based on the robot dynamic

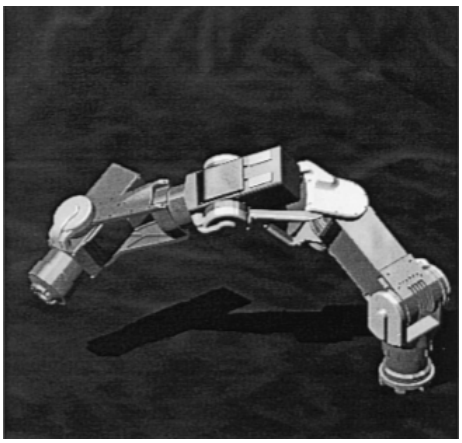


Figure 2.2: Robot used in the German Aerospace Center in 2003 for studies on human-robot interaction.

model together with the measured joints positions and motor torques.

A sensorless active LTP accomplished via a force control is presented in [26]. The controller is based on the external forces, that are estimated using the motor torques. The force estimation when the robot is not moving is a critical point in this approach, since the Coulomb uncertainties are particularly relevant at zero-velocity. A small dithering torque is here applied in feedforward when a joint is not rotating, in order to reduce the external torque required to start rotating a joint.

Another possible strategy to minimize the low velocity uncertainties, and therefore improve force estimation, is presented in [25]. In this paper, a passive LTP based on the feedforward compensation of gravity and friction is shown. The feedforward compensation is implemented by disabling the native low-level control loops; at the same time, the original controller is maintained active at low velocity, using a large integral gain, in order to have an helpful effect when a joint starts rotating. The above passive LTP has been implemented in the Robotics Lab at Lund University, on the same ABB YuMi robot used in this thesis. It was compared to the active LTP illustrated in this thesis, in order to highlight the improvements given by an active control; the results are shown in Section 6.3.3.

An example of active LTP is presented in [13], in which a force controller was designed for force feedback control, using the estimated external force. In this paper, two observers were implemented: the disturbance observer output estimator(DOOE), for the internal disturbances, and another one to include also the external forces. The former was tuned minimizing the difference between the two observers in absence of external

torques. The external force is then estimated comparing the output of the second disturbance observer and the DOOE, since the first one includes all the disturbances while the second one just the internal uncertainties.

Another recent sensorless LTP is illustrated in [21]. In this article, the authors aim to increase the accuracy of the system applying the force control only in one Cartesian direction at one time. The external torques are reconstructed, and a voting system identifies which is the largest Cartesian component, in order to obtain accurate lead-through programming in one direction of the operational space. The safety of the system is particularly considered: an optimization stage after the admittance controller establishes actuation bounds, safety-related limits on TCP velocity and avoidance of known obstacles. Furthermore, an additional algorithm is applied to avoid the collision of the whole kinematic chain of the robot with workspace objects.

2.4 Force Estimation Approaches

In order to implement an active LTP that does not require force sensors, force estimation algorithms are needed to detect the external forces.

Sensorless approaches usually involve disturbance observer algorithms to estimate the external forces applied by the operator, as shown in [29]. In this article the implementation of a Kalman filter in Cartesian space based on the generalized momentum is described. This approach is remarkably convenient since it does not require neither the inversion of the inertia matrix nor the calculation of the numerical acceleration, that are usually prone to numerical issues.

In [28] a filtered dynamic model for external force estimation is employed. It eliminates the need for acceleration measurements and uses a recursive least-squares estimation with exponential forgetting to estimate the force. Its main advantage is that, taking into account a configurable number of previous samples, allows the user to obtain a smooth output force signal, particularly useful if just noisy torque measurements are available. The presented algorithm turns out to be intimately connected to the disturbance observers based on the generalized momentum.

A common issue when using disturbance observers is the force estimation at zero-velocity. In fact, the uncertainties in the robot model related to joint friction are particularly large at low velocity, and this is an intrinsic limitation of model-based observers. Possible solutions to this problem aim to minimize this characteristic by reducing the friction torque at zero velocity, so to have a small initial estimation error.

Estimation algorithms based on the motor torques have usually very noisy estimates of the external forces, in particular at zero velocity. In [19] the authors are focused in force estimation during an assembly task, with particular attention on the problem of low velocity friction uncertainties. In this paper, the external force is estimated by solving a convex optimization problem in real-time, using prior knowledge of the contact force; an approximate confidence interval is also calculated.

All observer algorithms rely on the accuracy of the dynamic model of the mechanical system, and on the quality of the available measurements. Generally speaking, the resulting estimation of external forces might be less accurate of force/torque sensor measurements, but at the same time a sensorless implementation can be more reliable since it is not vulnerable to the typical sensor issues.

Chapter 3

Dynamic Model of ABB YuMi

In this chapter the methodology for the definition of the dynamic model of ABB YuMi is illustrated. A reliable dynamic model of the robot is essential to simulate the algorithms before the implementation of the real robot. Furthermore, the dynamic model of the manipulator will be used for the feedforward compensation of the gravity load on the joints during LTP, and for the implementation of the force observer, as it will be illustrated in Chapter 5.

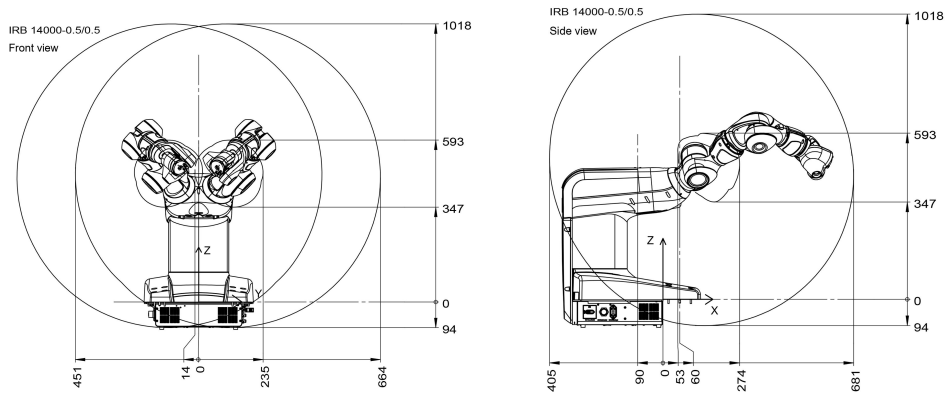
In the first section, the experimental set-up that was used in the Robotics Lab is described.

3.1 Experimental Set-up

All the experiments presented in this thesis were carried out in the Robotics Lab of the Automatic Control department at Lund University. In this section, a brief presentation of the robot and of the external research interface is given.

3.1.1 Robotic Manipulator

The ABB YuMi robot (also known as Frida) was used for all the experimental tests. It is a redundant dual-arm industrial robot where each arm has 7 degrees of freedom. It is designed for human-robot cooperative tasks, in fact, it has power and speed limitations and all the edges are covered with soft paddings [1].



(a) Front view of YuMi (b) Side view of YuMi

Figure 3.1: The ABB YuMi robot was used in the experiments.

3.1.2 Interface to the Robot

The native controller of the ABB YuMi is the ABB IRC5 control system. It controls each joint individually, with a main computer that calculates the references for all the joints. A cascaded control structure is used for each joint; in Figure 3.2 the joint model and the block diagram are shown. The outer position loop has a proportional gain, while the inner velocity loop has derivative and integral gains; there is also an inner control loop for the motor current, in order to ensure the desired motor torque.

In the Robotics lab, a research interface to interact with the original controller is available [8, 9]. This interface allows to set all the references of the low-level control (inputs) and to change the proportional, derivative and integral gain values of the PID controller; in addition, it is possible to read all the outputs. However, it is not possible to modify the control architecture. The low-level control loops run with a sampling frequency of 2kHz, while the research interface for setting the references and reading measurements runs at 250Hz.

The research interface is installed on an external PC where the OS Linux is used, while the Xenomai platform [4] is employed for real-time communication. The communication between the computers is realized using the LabComm protocol [3].

3.1.3 Force Sensor

The robot is equipped with a force-torque transducer ATI Mini40 [2], mounted on the end-effector flange, as shown in Figure 3.3.

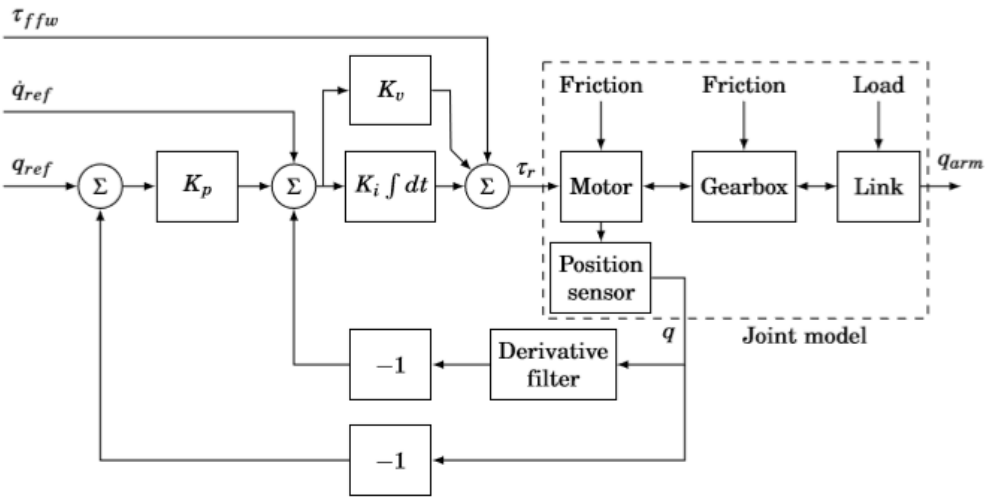


Figure 3.2: Block diagram of the low-level joint controller.

The sensor measures 3 Cartesian forces and 3 torques in the sensor coordinate frame. The raw measurements are then converted in the end-effector frame with a transformation matrix, scaled with the sensitivity of the sensor, and gravity compensated.

The measured forces in the end-effector frame are transformed in joint space, in order to know the torque on each joint:

$$\tau_{joint} = J_A^T F_{ee}$$

where F_{ee} is the 6-elements vector of the forces in the end-effector frame and J_A is the analytical Jacobian.

The sensor measurements are included in the external research interface, so that they can be used in the real-time routine as an input.

In this thesis, since the main goal was a sensorless LTP, the sensor measurements were employed just for validation and tuning purposes.

3.2 Kinematic Model

The first step to develop a model of an anthropomorphic manipulator is the definition of the kinematic chain of the arm.

A serial-link manipulator comprises a chain of mechanical links and joints; one end of the chain, the base, is generally fixed and the other end, the end-effector, is free to move. Each joint has one degree of freedom,

Figure 3.3: The force/torque sensor is mounted between the end-effector flange and the tool.

either translational (a sliding or prismatic joint) or rotational (a revolute joint).

A possible systematic way to describe the geometry of a serial chain of links and joints is the Denavit-Hartenberg notation (DH), proposed in 1955. An extended literature can be found regarding this convention, an example can be found in [23].

A link is considered a rigid body that defines the spatial relationship between two neighboring joint axes, and each link is characterized by its own coordinate frame. As shown in Figure 3.4, the relative position of each link is defined with respect to the previous one and in particular the *i-joint* will connect *link i-1* with *link i*.

It can be shown that, using the structure of kinematic chains, it is possible to define an homogeneous transform matrix between two coordinate frames by just 4 parameters:

1. $\theta_\mathbf{i}$: the link angle, between the x_{i-1} and x_i axes about the z_{i-1} axis.

2. $\mathbf{d_i}$: the link offset, the distance from the origin of frame $i-1$ to the x_i axis along the z_{i-1} axis.

3. $\mathbf{a_i}$: the constant link length, the distance between the z_{i-1} and z_i axes along the x_i axis.

4. $\alpha_\mathbf{i}$: the constant link twist, the angle from the z_{i-1} axis to the z_i axis about the x_i axis.

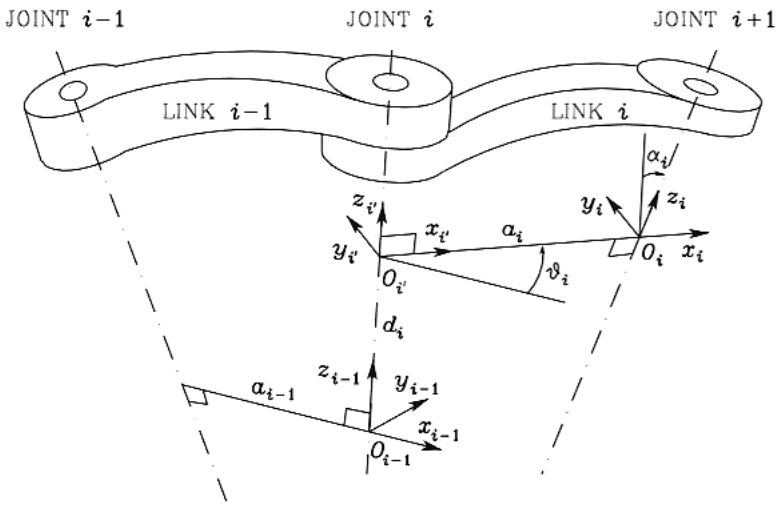


Figure 3.4: Denavit-Hartenberg parameters

According to the DH notation, two fundamental conditions have to be satisfied to define the kinematic chain of the manipulator:

1. The axis $\mathbf{x_i}$ is perpendicular to the axis $\mathbf{z_{i-1}}$
2. The axis $\mathbf{x_i}$ intersects the axis $\mathbf{z_{i-1}}$

In an industrial manipulator the base is usually fixed to the environment, while the final joint, *joint N*, connects *link N-1* to *link N*. The link N represents the tool of the robot and the parameters d_N and a_N specify the length of the tool and its x-axis offset respectively.

Once the DH parameters of each link are defined, it is possible to compute the homogeneous transformation from link coordinate frame *j-1* to frame *j*, that in terms of elementary rotations and translations is

$$A_j^{j-1} = T_{Rz}(\theta_j)T_z(d_j)T_x(a_j)T_{Rx}(\alpha_j) \tag{3.1}$$

where T represents the transformation matrix for the single rotation/ translation. The total homogeneous transformation A_j^{j-1} results written as

$$A_j^{j-1} = \begin{bmatrix} cos\theta_j & -sin\theta_j cos\alpha_j & sin\theta_j sin\alpha_j & a_j cos\theta_j \\ sin\theta_j & cos\theta_j cos\alpha_j & -cos\theta_j sin\alpha_j & a_j sin\theta_j \\ 0 & sin\alpha_j & cos\alpha_j & d_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

Once defined the homogeneous transform of each link, the *forward kinematics* can be computed to find the transformation from the base frame to the end-effector frame:

$$T_E^0 = A_1^0 A_2^1 \cdots A_N^{N-1}$$

Following the DH notation, the 4 parameters that describe the generic link have to be defined in a precise order: θ_i, d_i, a_i and α_i, for each link of the arm. If the joint is a revolute joint the variable θ will be replaced with the actual value of the angle of the joint, if the joint is a prismatic joint the variable a will be replaced with the actual length of the joint.

In the ABB YuMi all joints are of the revolute type. Nevertheless, in order to assign the DH parameters, it is not possible to place all the link frames where the real joints of the arm are, because the second condition of the DH notation would not be satisfied.

The kinematic chain of YuMi's arm is defined on the base on a reserved data-sheet by ABB with geometrical and inertial data. The arm configuration shown in Figure 3.5 is taken as reference, since the center of mass (COM) position of each link is referred to this pose.

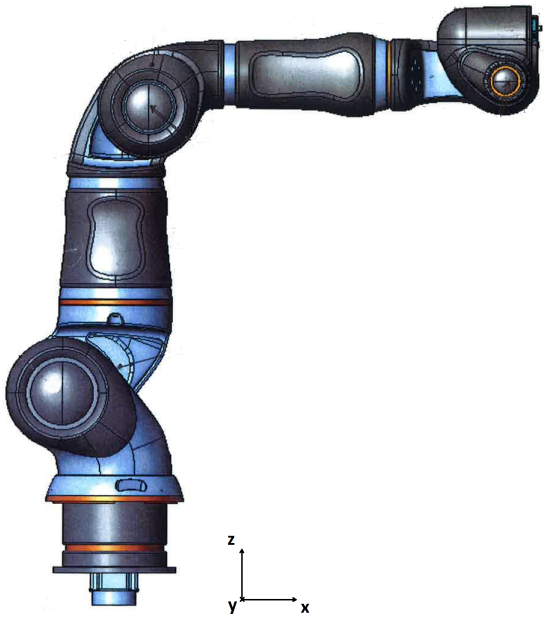


Figure 3.5: YuMi's arm: reference position for geometrical and inertial data

The ABB convention for joints names is different from an intuitive incremental classification; the following table matches the joints names starting from the base of the manipulator:

Incremental enum.	ABB enum.	Joint description
1	0	base
2	1	shoulder
3	6	redundant joint
4	2	elbow
5	3	wrist
6	4	wrist
7	5	end-effector

Table 3.1: YuMi's joints names: incremental enumeration and ABB notation.

For joints 3, 5 and 7 (in the incremental enumeration) is not possible to place the coordinate frame of the link where the real position of the joint is, since the second necessary condition is not satisfied: the axis $\mathbf{x_i}$ has to intersect the axis $\mathbf{z_{i-1}}$. The position of those frames in the model is then translated, so that the second DH condition is satisfied. As a result, the lengths of the links of the robot model are different from the real lengths, but the kinematics of the model is identical. This difference between the real and the modelled kinematic chain will have to be considered in the definition of the dynamic parameters.

In Figure 3.6 the base frame, the frames of the links and the positions of each center of mass (COM) are shown. It is important to notice that $frame_i$ defines $link_i$, and it is placed in correspondence of $joint_{i+1}$. The resulting kinematic model of YuMi's arm, implemented with Corke's toolbox in Matlab [11], is shown in Figure 3.7.

3.3 Dynamic Model

In order to define a dynamic model of the robot, the dynamic parameters of each link have to be known. All the dynamic parameters of each link have been given in a reserved ABB data-sheet: mass, center of mass (COM), inertia matrix defined at the COM and length of the link. In addiction, the gear ratios and motors inertia (high-speed side) have been provided. In particular, the position of the *i-th* centre of mass were given with respect to a frame with the same orientation of the base frame (x-forward, y-left, z-up) of figure 3.6, placed in the *i-th* joint. In order to express the coordinates of the COM in the link frame, as used in the DH notation, it is necessary to apply a rotation and a translation of the COM coordinates.

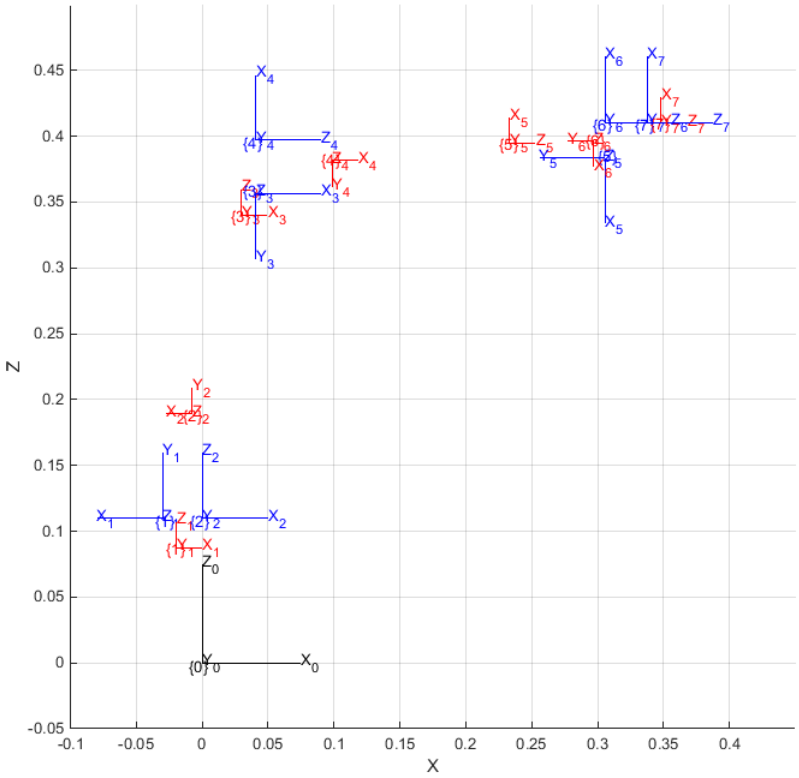


Figure 3.6: Base frame (black), link frames (blue), centers of mass (red)

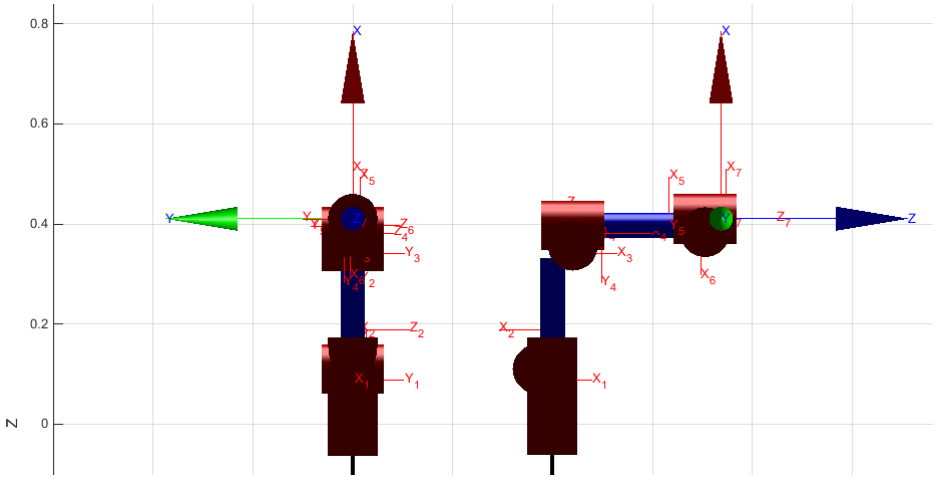


Figure 3.7: Model of YuMi's arm. The joints are drawn as red cylinders, links as thin blue cylinders, the red frames indicate the position of COMs, the final frame indicates the end-effector

The inertia matrix of each link is already defined in correspondence of the COM, so that, to express the inertia matrix in the link frame, the matrix given in the data-sheet has to be rotated but not translated.

As an example, if X_{BF} denotes the COM coordinates vector with respect to the base frame, the COM coordinates with respect to the link frame will result:

$$X_{LF} = T_{LF}^{BF} R_{LF}^{BF} X_{BF} \tag{3.3}$$

where T_{LF}^{BF} and R_{LF}^{BF} are the translation and rotation matrices from the base frame to the link frame.

All the properties of each link are assigned using the respective class in Peter Corke's Matlab toolbox [11]. The Matlab code for the definition of the dynamic model is reported in Appendix B.

3.4 Kinematic Validation of the Model

In order to have a faithful kinematic model of the arm, once the kinematic chain of the manipulator is defined, it is necessary to determine the base orientation and translation, and the offsets of joint angles with respect to the encoders references in the real robot. The orientation and translation of the base with respect to the world frame were provided.

The angle offsets were found with a two-steps process. Firstly, they have been roughly identified visually, so as to have a starting value for the numerical optimization. The exact offset values were identified minimizing the error between experimental and simulated end-effector positions. In particular, the experimental end-effector positions were computed with a verified forward kinematic function, taking the measured joint positions as input. Using the same joint position measurements, the simulated end-effector trajectory were computed using the kinematic model described above. The final offsets were found using the gradient method, minimizing the squared difference between the real Cartesian coordinates and the simulated ones.

Figure 3.8 shows a test where the robot was moved by jogging through the teach pendant. The end-effector accomplishes a wide trajectory that covers a large part of the operational space of the arm. The data of the Cartesian coordinates of the end-effector were logged and then compared with the ones calculated using the model, given the joints angles. It can be observed a perfect matching between real and simulated coordinates, that proves the precision of the kinematic model.

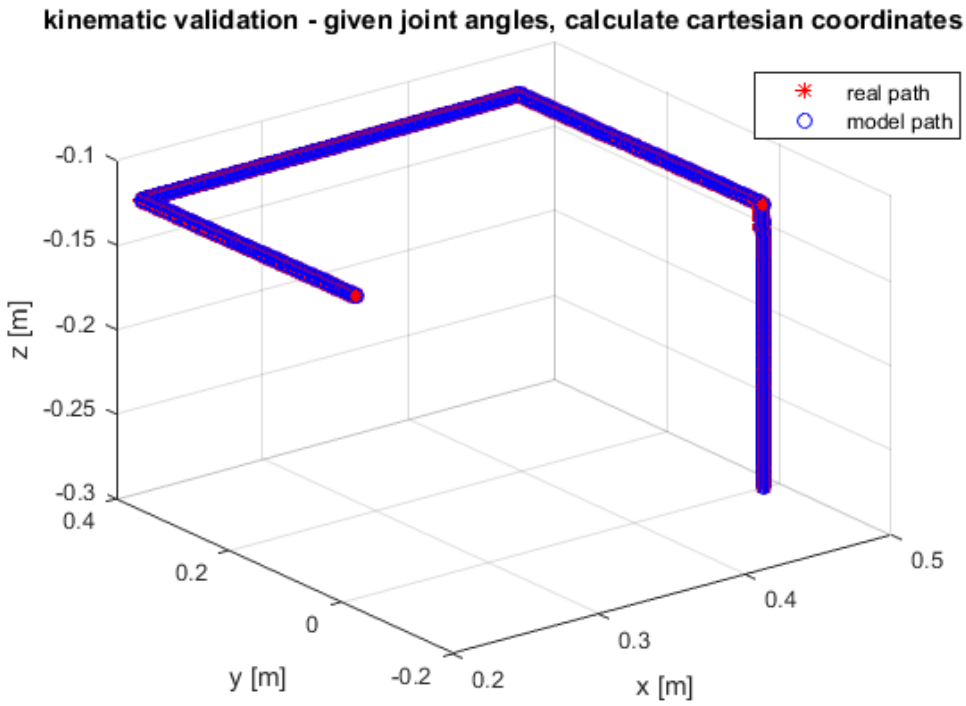


Figure 3.8: Test for the kinematic validation of the robot model. The measured Cartesian coordinates of the end-effector were compared to the modelled ones, given the measured joint angles. It can be noticed a perfect matching between the real (measured) and the simulated Cartesian paths.

Chapter 4

Passive LTP

The first phase of this thesis project involved the implementation of a passive lead-through programming, based on the feedforward compensation of the gravity load and part of joint friction. A similar passive LTP has been illustrated in [25].

The feedforward compensation of gravity and friction is model based, therefore, an accurate dynamic model of the robot is required. In Chapter 3, the kinematic model and the physical properties of the robot links have been defined, on the basis of the provided data sheets. As concerns the definition of the joint friction model, a long campaign of experiments was necessary. In the next section, the resulting friction model is described.

4.1 Joint Friction Model

A reliable joints friction model is fundamental in order to obtain satisfying performances in friction compensation and external torque estimation. In fact, a friction model is required to calculate the feedforward compensation torque for the implementation of a passive LTP. Furthermore, as it will be explained in Section 5, it is essential in the force observer algorithm for the correct estimation of the external torques. In this section, the methodology for friction identification and the friction model are presented.

4.1.1 Identification Experiment

Experimental tests were performed on each joint of YuMi to identify the unknown friction parameters.

The joint was moved back and forth setting an acceleration reference

with a squared wave profile, as shown in Figure 4.1. The reference was chosen so to obtain a triangular velocity profile, i.e. a linear velocity trend. The amplitude of the velocity reference was set equal to 0.2 rad/s, that is not so different from the average joint velocity during LTP, where usually the handling of the robot is accomplished with slow movements. In order to neglect secondary dynamic effects, the angular acceleration of the joint has to be small; for this reason, it is convenient to choose a low frequency of the desired motion law. Finally, a frequency of 0.1 Hz and an amplitude of 0.2 rad/s have been chosen for the velocity reference.

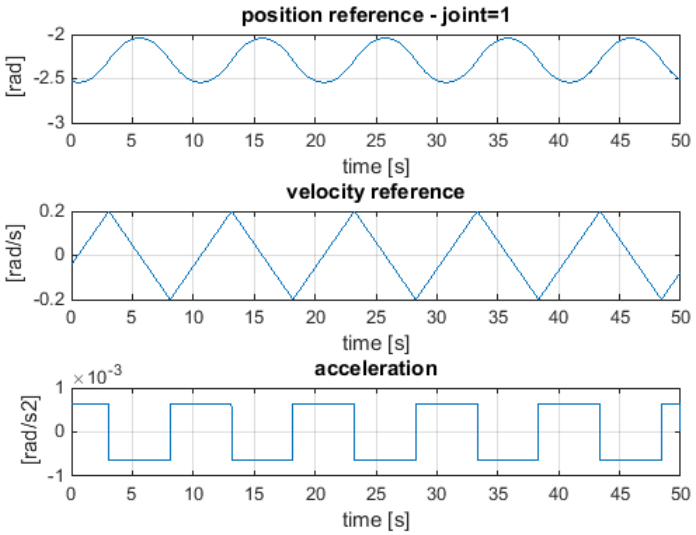


Figure 4.1: Reference for position, velocity and acceleration in the test for friction identification

The main outcome of this experiment is a velocity-torque curve, shown in Figure 4.2, useful for the evaluation of the friction model parameters. The raw data, sampled at 250 Hz, are low-pass filtered with the discrete-time filter (4.1) to remove some of the noise.

$$H(z) = \frac{0.4}{1 - 0.6z^{-1}} \tag{4.1}$$

It can be seen that the blue curve in Figure 4.2 is not symmetrical with respect to x-axis, since part of the motor torque is due to counteract the gravity component; furthermore, it presents high measurement noise. For this reason, it can be noticed that the magnitude of the torque for positive increasing velocity differs from the magnitude related to positive

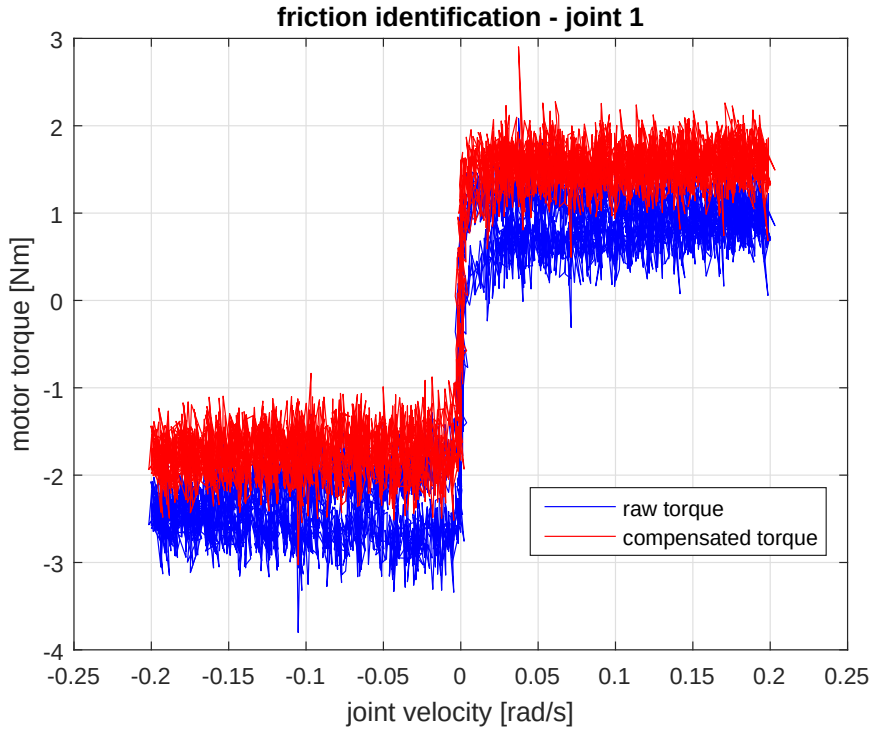


Figure 4.2: Curve obtained from friction identification experiment. The raw torque is plotted in blue; it is not symmetrical with respect to the x-axis because of the gravity load on the joint. The red line is obtained after filtering the velocity and torque measurements and compensating the gravity component.

decreasing velocity, especially at velocities close to zero, where the position of the joint reaches the extremities of the imposed displacement. Applying a gravity compensation on the raw torque and taking the filtered velocity instead of the raw velocity, the resulting curve is plotted in red in Figure 4.2. This curve has a lower noise and is centered with respect to the x-axis. It shows a typical S-shape trend, due to the combination of viscous and Coulomb friction. The friction magnitude at zero velocity varies between the maximum and the minimum value of the Coulomb component and, due to the noise in the velocity measurements, this is practically true also for measured velocities close to zero. The viscous component is also visible in the curve, since the magnitude of the measured motor torque increases at high velocities.

This experiment was performed at low and high angular velocities, for each joint of YuMi's arm.

Joint num.	Description	τ_C [Nm]	A	B	c
1	base	1.5	5e+03	2e-03	1.1
2	shoulder	1.4	5e+03	2e-03	0.9
3	redundant	0.85	5e+03	2e-03	0.7
4	elbow	0.7	5e+03	2e-03	0.7
5	wrist	0.25	5e+03	3e-03	0.05
6	wrist	0.4	5e+03	2e-03	0.1
7	e-e	0.24	5e+03	2e-03	0.03

Table 4.1: Friction model parameters: these parameters were manually tuned in order to make the sigmoid functions fit the measured friction trend.

4.1.2 Friction as a Probabilistic Variable

Joint friction can be described as a known disturbance, in order to be taken into account during the external force estimation. Therefore, the characteristics of the disturbance have to be identified. A similar friction identification experiment has been carried out in [24].

The S-shaped trend of friction torque, shown in Figure 4.2, can be well captured by two sigmoid functions defined as:

$$\tau_{f,max}(\dot{q}) = \tau_{C,min} + \frac{\tau_{C,max} - \tau_{C,min}}{1 + e^{-A(\dot{q}+B)}} + c\dot{q} \tag{4.2}$$

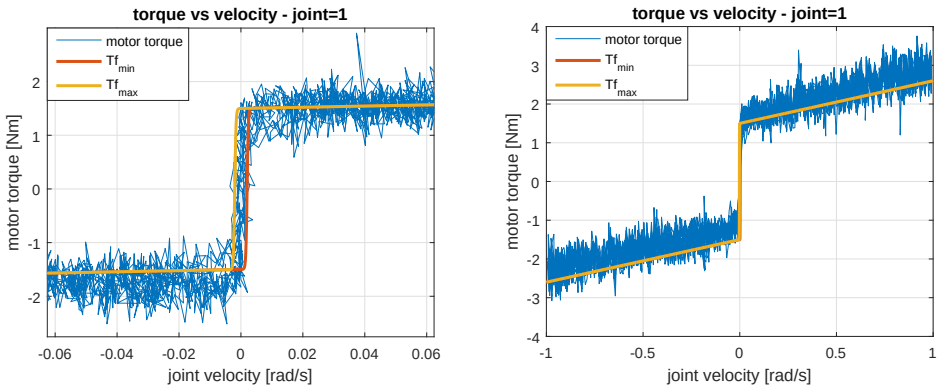
$$\tau_{f,min}(\dot{q}) = \tau_{C,min} + \frac{\tau_{C,max} - \tau_{C,min}}{1 + e^{-A(\dot{q}-B)}} + c\dot{q} \tag{4.3}$$

where τ_C represents the Coulomb friction, the parameter A describes the slope of the sigmoid function, the parameter B the width of the area between the curves and finally a linear term $c\dot{q}$ models the viscous component.

The friction identification experiment was performed for each joint of the manipulator, and the parameters of the sigmoid functions were manually tuned for each joint. The resulting experimental parameters are reported in Table 4.1.

An example can be seen in Figure 4.3, where the sigmoid functions are tuned in order to fit the measured trend of the friction torque.

The friction torque, due to the combination of Coulomb and viscous friction, has been modelled as a velocity-dependent Gaussian random variable. With this model the friction torque can be described as a Gaussian variable, i.e. a variable with a mean value and a variance that is equivalent



(a) low angular velocity (b) high angular velocity

Figure 4.3: Tuning of the sigmoid functions. The parameters of the two sigmoid functions (red and yellow lines) are tuned so that the functions can fit the measured trend of the friction torque.

to the uncertainty. The mean value of the probabilistic variable can be set equal to the average of the two sigmoid functions defined in Equation (4.2), calculated as

$$\bar{\tau}_f(\dot{q}) = \frac{\tau_{f,max}(\dot{q}) + \tau_{f,min}(\dot{q})}{2} \tag{4.4}$$

where the subscript f refers to friction.

The variance of the Gaussian variable represents the friction uncertainty. It was defined as a velocity-dependent uncertainty in which two components can be identified: a constant term with the velocity, due to measurement noise and irregularities in the joint, and a second term due to the impossibility to exactly know the friction torque when the joint is not moving (static friction uncertainty).

The first component of the friction uncertainty was experimentally identified with the test presented in the previous section. It was set equal to the variance of the friction torque for high positive velocities, and it is considered constant in all the velocity range.

The second component of uncertainty describes the impossibility to exactly determine the real Coulomb friction value at almost-zero velocity. In fact, in a small velocity range around zero, the magnitude of the friction can take any value between the minimum and the maximum of the Coulomb component. This uncertainty can be described combining the

two sigmoid function with the following formula:

$$v_C = \frac{\tau_{f,max}(\dot{q}) - \tau_{f,min}(\dot{q})}{2} \tag{4.5}$$

where the subscript C refers to the Coulomb friction. The resulting uncertainty v_C is shown in Figure 4.4.

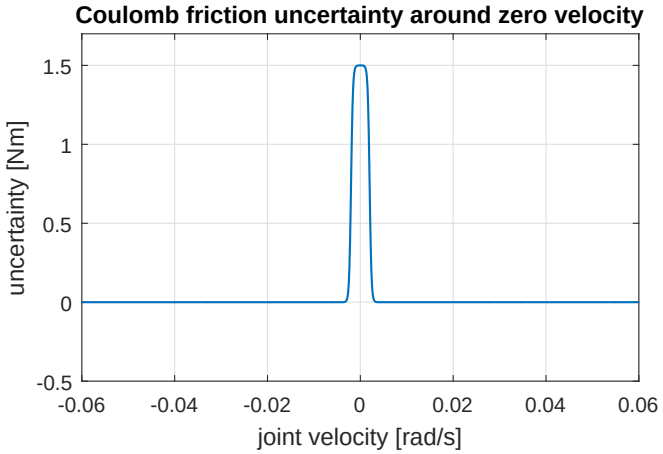


Figure 4.4: Qualitative trend of the Coulomb friction uncertainty around zero velocity.

In conclusion, the resulting friction model is characterized by a velocity-dependent Gaussian variable, thus, by a mean value and a correspondent uncertainty. The resulting model is illustrated in Figure 4.5. It can be noticed that the friction around zero velocity is described with mean value equal to zero and a very large uncertainty. As soon as the joint starts rotating the velocity increases and the friction can be described with more accuracy, therefore the average value is not zero and the correspondent uncertainty is lower.

Describing the friction torque with this approach is particularly convenient since it can be successively employed in the implementation of the Kalman filter for force estimation. In fact, the Kalman filter approach requires the definition of the robot model uncertainty; since the friction uncertainty is the major component, the model uncertainty will be set equal to the friction uncertainty. This idea will be further developed in Chapter 5.

4.1.3 Non-modeled Disturbances

As already well known, in a real system, friction is often the most difficult part to model correctly. In an industrial manipulator, the joint friction

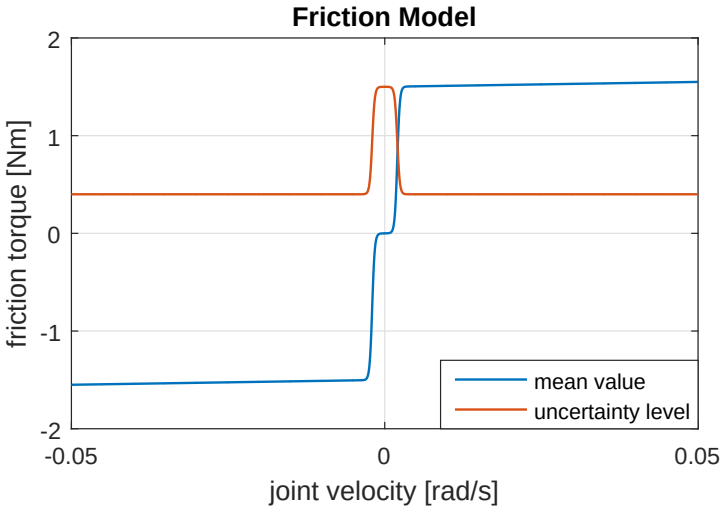


Figure 4.5: Friction model: velocity-dependent Gaussian random variable. The mean value (blue line) and the corresponding variance (red line) are shown.

can be influenced by countless factor, e.g. the operative conditions (temperature, humidity) and the internal characteristics of the manipulator, such as the manufacturing quality and the use.

In the YuMi robot used for the experiments, a position-dependent friction torque can be observed. In particular, this phenomenon can be noticed in the test for friction estimation illustrated in Section 4.1.1 (back and forth movement with a square-wave acceleration), especially repeating the test setting high velocities and large displacement. In Figure 4.6, the velocity-torque curve obtained for high velocities and large displacement is shown. When the joint reaches the extreme positions of the joint rotation, the magnitude of the motor torque that counteracts friction (gravity and inertia components have been compensated as explained before) is significantly different. In fact, the extreme positions of the imposed displacement are reached when the velocity is zero; the joint friction at zero velocity should theoretically be the same, independently from the angular position, but, in the test shown in Figure 4.6, this hypothesis is not verified (the black circles in the picture indicate the different magnitude of the Coulomb friction at almost-zero velocity). This unexpected behavior is possibly due to the internal cables that run through the link that interfere with the link rotation, in addition with others small structural issues. As a result, the friction torque depends on the position and the velocity of the joint.

Since the position dependency was not modelled, the resulting varia-

tion of the friction torque was considered as an unmodelled disturbance.

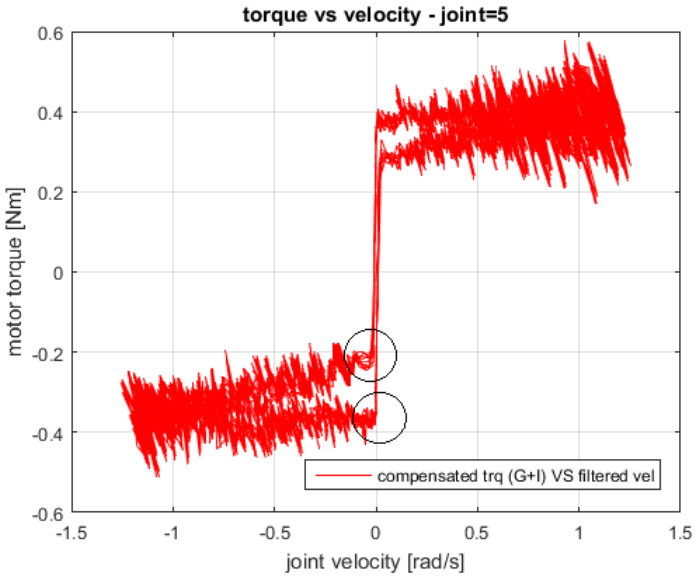


Figure 4.6: Position-dependency of the friction torque. It can be seen that the motor torque due to friction is significantly different in the extreme positions of the imposed displacement (black circles).

4.2 Feedforward Compensation

In this section, the feedforward compensation of gravity and friction to accomplish a passive LTP is illustrated.

In order to manually move the robot arm, the original PID controller implemented by ABB was deactivated, so that no control loops were active to maintain the robot stiff in the current position. To disable the ABB control system, the proportional, derivative and integral gains were set to zero, using the external research interface [8]. However, the resulting control torque on the joint is not zero, because of the remaining constant integral part of the PID controller. To take into account and compensate this undesired torque, the last value of the control torque was stored when the PID controller was disabled. Then, this value was subtracted to the feedforward gravity compensation in order to obtain the desired motor torque.

4.2.1 Gravity Compensation

In order to calculate the gravity load on each joint, a recursive Newton-Euler method (RNE) based on the dynamic model of the robot was implemented [11].

The RNE function to calculate the gravity load on each joint was coded adapting Corke's function called *rnedh* [11], for models defined using standard DH notation. The resulting Matlab function can be compiled on the robot controller and recalled during the real-time routine. The function takes the joint positions as input and calculates the gravity torque using kinematic and dynamic parameters, by a recursive forward and backward method. The Matlab function is reported in Appendix A. The same function can also calculate the inertia matrix of the robot, that was used to compute the generalized momentum, as illustrated in Section 5.

The calculated gravity load was applied as a feedforward torque. The resulting motor torque appeared to be sufficient to maintain the manipulator in a fixed position without external forces, no drifts were observed. It has to be noticed that, in this case, the joint friction has a positive influence to keep the robot still when no external forces are applied, since it compensates the model uncertainties. As a result, the robot arm can be moved by hand and it stands if no external forces are applied. However, the handling of the arm is not really comfortable and, because of the joint friction, a considerable force is needed to move the joints.

4.2.2 Friction Compensation

When the native robot controller is disabled and the gravity load is compensated, the joint friction torque is helpful as it prevents the robot from moving, except the case in which external forces are applied. On the other hand, a feedforward friction compensation is needed in order to make the robot easily movable during the passive LTP. Since friction is generally very difficult to model accurately, the friction model is characterized by an high uncertainty level. Therefore, only a percentage of the modelled friction was compensated, so that no undesired joints drift were observed. In fact, there should still exist a friction torque that stops the robot if no forces are applied; the amount that can be compensated depends on the viscous friction level and on the accuracy of the model. The friction compensation torque was defined on the bases of the friction model presented in Section 4.1.

The sign of the Coulomb friction depends on the direction of rotation of the joint, therefore on the velocity measurements. Since the velocity mea-

surements present white noise, when the joint is not rotating the measured velocity is not exactly zero. For this reason, friction can be compensated once the velocity exceeds the measurement noise level, and this threshold can be experimentally evaluated. Furthermore, the amount of friction compensation was made proportional to the velocity for small velocities, in order to smooth the transition between zero and total compensation. Similarly to [25], the friction compensation torque was defined as

$$\tau_f = \begin{cases} 0 & , |\dot{q}| < \dot{q}_0 \\ \frac{|\dot{q}|-\dot{q}_0}{\dot{q}_1-\dot{q}_0}\text{sign}(\dot{q})\epsilon\tau_C & , \dot{q}_0 \leq |\dot{q}| < \dot{q}_1 \\ \text{sign}(\dot{q})\epsilon\tau_C & , |\dot{q}| \geq \dot{q}_1 \end{cases}$$

where $\dot{q}_0$ and $\dot{q}_1$ are the velocity thresholds of the linear compensation with the velocity, and ϵ is the percentage of the modelled Coulomb friction that was compensated. The qualitative friction compensation trend is shown in Figure 4.7.

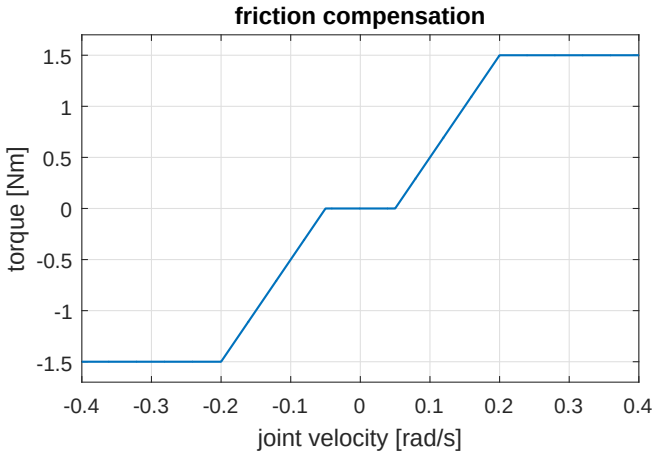


Figure 4.7: Example of the friction compensation torque

The threshold value $\dot{q}_0$ was experimentally set equal to $\dot{q}_0 = 0.003$ rad/s, in order to be greater than the velocity measurement noise.

The max percentage ϵ of compensated Coulomb friction was set equal to 70%.

In order to smooth the feedforward friction compensation torque, the low-pass filter defined in Equation (4.6) was added, so that the user can easily move the arm without perceiving an abrupt change when the joints start moving.

$$H(z) = \frac{0.4}{1 - 0.6z^{-1}} \tag{4.6}$$

4.3 Passive LTP Experiment

The passive lead-through programming was implemented disabling the native ABB controller and applying the feedforward compensation of gravity and friction illustrated in the previous section. The algorithm was programmed in Matlab and then compiled on the real-time robot controller using the external research interface [8].

Thanks to the passive LTP, YuMi's arm can be moved by hand applying a small external force. It was observed that a major force is required to start rotating a joint, due to the static friction that it is not compensated. Furthermore, moving the end-effector while keeping its orientation is not a very comfortable task, since it requires the simultaneous rotations of several joints, therefore a major external torque has to be applied.

In Figure 4.8 the feedforward motor torque applied during the passive LTP is shown. The feedforward components related to the gravity load and friction can be distinguished. In particular, it can be noticed that the friction compensation is active when the joint is rotating, i.e. if the angular velocity differs from zero. As soon as the joint starts rotating and the angular velocity exceeds a chosen threshold, the friction torque is smoothly compensated.

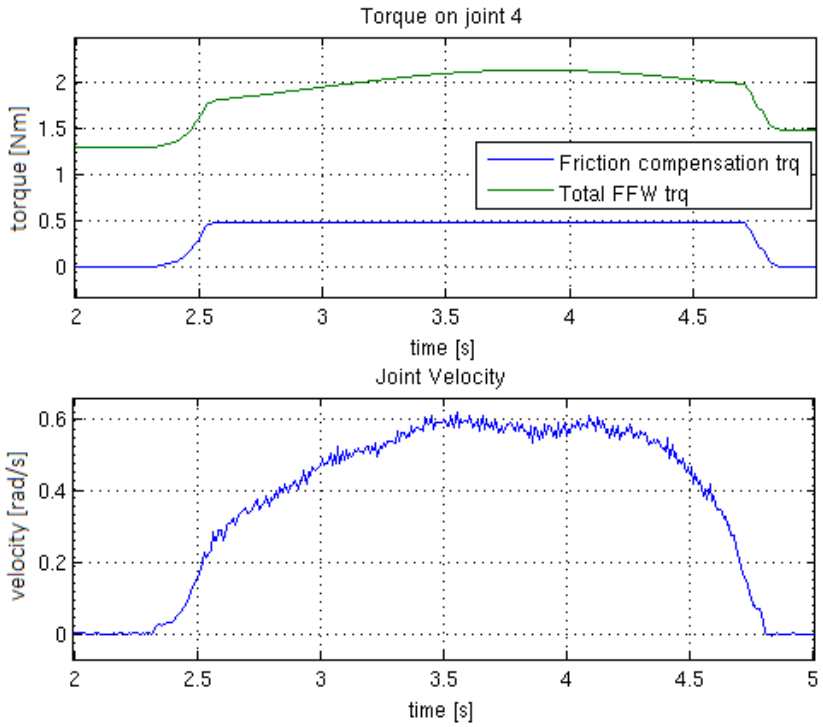


Figure 4.8: Feedforward torque to compensate the gravity and the friction. It can be noticed that the friction compensation is active only when the joint is moving.

Chapter 5

Disturbance Observer

In order to implement a lead-through programming supported by a force control, the external forces applied by the user have to be known. In this thesis, a sensorless approach has been adopted, therefore, the external forces have been estimated by a force observer. In this chapter, a Kalman filter based on the generalized momentum formulation is presented. It has been implemented on the ABB robot YuMi; in the next sections, the tuning experiments and the performance of the observer are illustrated.

5.1 Kalman Filter based on the Generalized Momentum

Existing force estimation schemes relying directly on the robot dynamics typically involve computation of joint accelerations or inversion of the inertia matrix. The former approach requires numerical differentiation of joint speeds going along with the amplification of measurement noise, while the latter may be computationally costly and prone to numerical issues. As a solution, a force observer in joint space based on the generalized momentum approach is presented. As opposed to the previous formulations, it does not require neither the numerical acceleration nor the inversion of the inertia matrix [29].

The equations of motion of a manipulator with n degrees of freedom can be written as [22]:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_{fric}(\dot{q},q) + \tau_{ext} = \tau \tag{5.1}$$

where $q \in \mathbb{R}^n$ denotes the joint angles, $\tau_{ext} \in \mathbb{R}^n$ are the external torques on joints, $M \in \mathbb{R}^{n\times n}$ denotes the inertia matrix, $C \in \mathbb{R}^{n\times n}$ the Coriolis

matrix, $G \in \mathbb{R}^{n\times n}$ the gravity load, $\tau_{fric} \in \mathbb{R}^n$ the friction torque and $\tau \in \mathbb{R}^n$ the motor torque on the arm side of the gearbox.

The following approach has been presented by Wahrburg et al. in [29], in this thesis it will be elaborated and enriched.

The generalized momentum is given by

$$p = M(q)\dot{q} \tag{5.2}$$

and the differentiation with respect to time results in

$$\dot{p} = \dot{M}(q)\dot{q} + M(q)\ddot{q} \tag{5.3}$$

Substituting (5.3) into (5.1) results in

$$\dot{p} = \dot{M}(q)\dot{q} + \tau - C(q,\dot{q})\dot{q} - G(q) - \tau_{fric}(\dot{q},q) - \tau_{ext} \tag{5.4}$$

Assuming that the Coriolis matrix is expressed using Christoffel symbols [23], $(\dot{M} - 2C)$ is a skew-symmetric matrix and, with symmetry of M, this implies

$$\dot{M} = C + C^T$$

Accordingly, Equation (5.4) can be further simplified to

$$\dot{p} = C(q,\dot{q})^T\dot{q} - G(q) + \tau - \tau_{fric}(\dot{q},q) - \tau_{ext} \tag{5.5}$$

The key idea is to combine the description of the manipulator dynamics based on the generalized momentum with well-known disturbance observer approaches [29]. To this purpose, the external torques can be modeled as

$$\dot{\tau}_{ext} = A_\tau\tau_{ext} + w_\tau \tag{5.6}$$

where w_τ is the Gaussian white noise (or uncertainty), with intensity Q_τ; the subscript τ indicates that it is related to the external torque.

Focusing on estimating external torques, a joint friction estimate $\hat{\tau}_{fric}$ is assumed to be available, in which the uncertainties in friction estimates are modelled as white noise w_m with intensity equal to Q_m

$$w_m = \hat{\tau}_{fric} - \tau_{fric} \tag{5.7}$$

where the subscript m stands for the model, since friction is the major source of uncertainties in the dynamic model of the robot. The term $\hat{\tau}_{fric}$ and the white noise w_m are determined using the friction model defined in

Section 4.1. Since joint friction has been modelled as a Gaussian velocity-dependent variable, $\hat{\tau}_{fric}$ was set equal to the mean value and w_m equal to the friction uncertainty.

Taking into account Equation (5.7) in Equation (5.5), the generalized momentum dynamics can be expressed as

$$\dot{p} = \tau + C(q,\dot{q})^T\dot{q} - G(q) - \tau_{ext} - \hat{\tau}_{fric}(\dot{q},q) + w_m \tag{5.8}$$
$$= u - \tau_{ext} + w_m \tag{5.9}$$

where the term u is defined as

$$u := \tau + C(q,\dot{q})^T\dot{q} - G(q) - \hat{\tau}_{fric}(\dot{q},q) \tag{5.10}$$

The above equation can be reformulated in the state-space form by augmenting the states with the external forces in accordance with Equation (5.6) and considering the term u as the input of the system:

$$\begin{cases} \overbrace{\begin{bmatrix}\dot{p}\\ \dot{\tau}_{ext}\end{bmatrix}}^{\dot{x}} & = \overbrace{\begin{bmatrix}0_n & -I_n\\ 0_n & A_\tau\end{bmatrix}}^{A}\overbrace{\begin{bmatrix}p\\ \tau_{ext}\end{bmatrix}}^{x} + \overbrace{\begin{bmatrix}I_n\\ 0_n\end{bmatrix}}^{B}u + \overbrace{\begin{bmatrix}w_m\\ w_\tau\end{bmatrix}}^{w} \\ y & = \underbrace{\begin{bmatrix}I_n & 0_n\end{bmatrix}}_{C}\begin{bmatrix}p\\ \tau_{ext}\end{bmatrix} + v. \end{cases} \tag{5.11}$$

Matrix A_τ determines the dynamics of the external wrench; assuming a piecewise constant torque, matrix A_τ can be chosen equal to zero. The term v describes the measurement white noise with intensity R. In a standard form the above system becomes

$$\begin{cases} \dot{x} &= Ax + Bu + w \\ y &= Cx + v \end{cases} \tag{5.12}$$

where matrices A, B, C are constant.

The corresponding observer to system (5.11) is

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + K(y - \hat{y}) \\ \hat{y} = C\hat{x} \end{cases} \tag{5.13}$$

where K is the gain matrix that multiplies the estimation error. This matrix can be computed using several observer algorithms. As already

said, in this thesis an optimal observer is adopted, therefore the K matrix is calculated as

$$K = PC^T R^{-1}, \tag{5.14}$$

Matrix P is the solution of the algebraic Riccati equation (ARE) [17]

$$AP + PA^T - PC^T R^{-1} CP + Q = 0. \tag{5.15}$$

In the equation above, matrices R and Q are the intensities of the measurements noise and the state noise respectively. Q is composed by the model uncertainty Q_m and the external torque uncertainty Q_{tau}:

$$Q = \text{blockdiag}([Q_m, Q_{tau}]).$$

Since the system matrices are constant and the noise intensities are not changing over a large range, the ARE, instead of the differential Riccati equation (DRE), can be used to solve for the Kalman gain K, approximating the solution with the stationary value.

Finally, the estimate of external torques on joints can be obtained from the estimated state $\hat{x}$. It is important to notice that the magnitude of the estimated torque will directly depend on the friction model defined for the observer, since there are no available measurements for the external forces.

Considering the uncertainties as velocity-dependent would mean that the Kalman gain matrix has to be calculated at every instant, by solving the ARE at each iteration of the real-time controller.

In Section 4.1 the experimental trend of the friction velocity-dependent uncertainty was illustrated. In particular, in Figure 4.5 it was shown that the low-velocity friction uncertainty turns out to be remarkably higher than the high-velocity uncertainty. For the sake of simplicity, the trend of the friction velocity-dependent uncertainty, shown in Figure 4.5, can be approximated by two levels of uncertainty: a constant high value for low velocities and a low value for velocities higher than a defined threshold. Therefore, two Kalman gain values can be defined for each joint; these values can be calculated off-line and then used during the real-time routine. In particular, recalling the observer system defined in Equation (5.13):

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + K(y - \hat{y}) \\ \hat{y} = C\hat{x} \end{cases}$$

the last term can be written as

$$K(y - \hat{y}) = \begin{bmatrix} Kp_{(N\times N)} \\ K\tau_{(N_f \times N)} \end{bmatrix} \left(\begin{bmatrix} p_{mes} \end{bmatrix} - \begin{bmatrix} p_{est} \end{bmatrix} \right) \tag{5.16}$$

The uncertainties related to noise measures and to external torques are assumed to be constant.

Matrices Q and R are defined as diagonal. Solving the ARE, the resulting Kalman gain matrix is composed by two diagonal matrices, Kp and $K\tau$, the first one associated with the generalized momentum while the second one with the external torques (since the state of the observer is composed by the generalized momentum and the external force). The upper part of the Kalman gain matrix, matrix Kp, will be so redefined at every step of the discrete time controller: each component on the diagonal (associated with the related joint) can be chosen between two gain values, depending on the velocity of the related joint. For instance, if the first joint of the robot is not moving, the first value on the diagonal of Kp will set equal to the high uncertainty value corresponding to the first joint. If the second joint is moving, the lower value between the two uncertainty levels associated with the second joint will be chosen and inserted in the second position of the diagonal.

Once the Kalman gain matrix is defined, the input of the observer u can be calculated as explained in Equation (5.5), and finally the state of the system can be estimated. The resulting block diagram is shown in Figure 5.1.

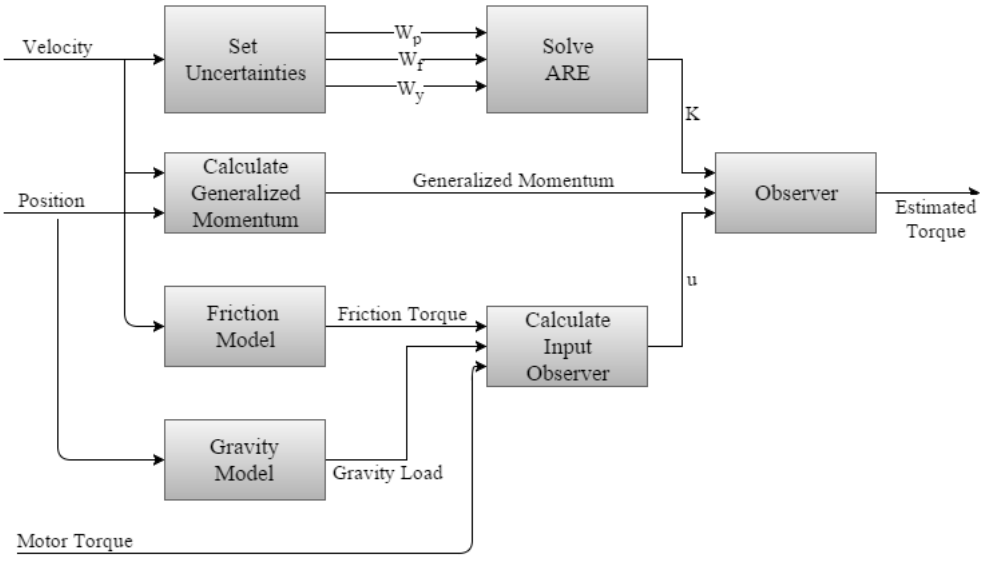


Figure 5.1: Block diagram of the force observer.

As can be seen in Equation (5.13), the inputs of the observer are the artificial torque u, defined in Equation (5.10), and the measured generalized momentum y. It is clear that if the measured joint velocity is null

both y and $\hat{\tau}_{fric}$ are zero. An intrinsic limitation of the Kalman filter based on the generalized momentum approach is that the external torques can be estimated only if the joints are moving. Therefore, only the external forces that overcome the joint static friction can be estimated. In the next sections, this property of the observer is experimentally verified, and a strategy that aims to mitigate this undesired aspect is presented.

5.2 Implementation and Tests

The Kalman filter was implemented in Simulink and then compiled on the real-time robot controller using the external research interface [8]. The block diagram of the force observer is illustrated in Figure 5.1. In particular, the friction model block calculates the friction torque using the model described in Section 4.1. The gravity model block computes the gravity load on joints using the RNE algorithm illustrated in Appendix A.

In this section, the experiments concerning the tuning of the observer are described. The force sensor mounted on the end-effector, described in Section 3.1.3, was used just for validation and tuning purposes.

5.2.1 Tuning of the Observer

The Kalman filter illustrated in the previous section was experimentally tuned in order to estimate the external forces correctly. In particular, the white noise of the generalized momentum measurements, the uncertainty of the dynamic model and the uncertainty referred to the external forces have to be defined, so that the Riccati equation (5.15) can be solved.

The intensity of the noise of angular velocity measurements (arm side) is experimentally estimated equal to 10^{-4} [rad/s]. Since the observer works with the generalized momentum, the noise level associated with the measured generalized momentum is set equal to 10^{-6} [kg rad/s]. Accordingly, the corresponding uncertainty matrix R is defined as

$$R = 10^{-6} I_n \tag{5.17}$$

where I_n is $n \times n$ identity matrix.

Matrix Q is composed as

$$Q = \text{blockdiag}([Q_m, Q_\tau]).$$

where Q_m and Q_τ refer to the dynamic model and the external torques uncertainties respectively.

The dynamic model uncertainty is basically due to friction uncertainty. As already said, the friction uncertainty can be simplified into a two-levels uncertainty, one at almost-zero velocities and the second one at higher velocities; the friction uncertainty was shown in Figure 4.5. Therefore, two values of uncertainty for each joint can be defined, depending on the joint velocity. As a result, the diagonal matrix Q_m was defined at every step of the real-time controller: depending on the joint velocity, the n-th component of the Q_m matrix was chosen between the two uncertainty values of the correspondent joint.

The choice of the external torque uncertainty Q_τ determines the estimation speed of the external torque: a high level will lead to an abrupt step in the estimation with large overshoots, on the contrary, a too small uncertainty will let the observer trust less the measurements resulting in a "lazy" estimate. A trade-off value is chosen experimentally, equal to 10. The resulting uncertainty matrix was defined as

$$Q_\tau = 10 I_n \tag{5.18}$$

where I_n is $n \times n$ identity matrix.

It is important to notice that the magnitude of the estimated torque will entirely depend on the friction model, since there are not available measurements for the external forces.

In order to test and tune the force observer, the passive LTP illustrated in Section 4.2 was used, i.e. the gravity and friction torques were compensated with a feedforward control.

In the experiments, the estimated external torques were compared to the measured torque in joint space. The external torques are measured with the force sensor mounted on the end-effector, as illustrated in Section 3.1.3.

In Figure 5.2, a test in which an external torque was applied on a joint of YuMi is illustrated. It can be noticed that, as soon as the joint starts rotating, the estimated torque converges to the measured one. On the other hand, the torque estimate clearly shows that the observer cannot detect the external torque if the angular joint velocity is equal to zero. In fact, the inputs of the observer are the artificial torque u, defined in Equation (5.10), and the measured generalized momentum y; it is clear that if the measured joint velocity is null the resulting estimated torque will be zero, since both the inputs of the observer are constant or zero. Therefore, a limit of the Kalman observer based on the generalized momentum is that the external torques can be estimated only if the joints are moving, which means that just the forces that overcome the static friction in the joints can be estimated.

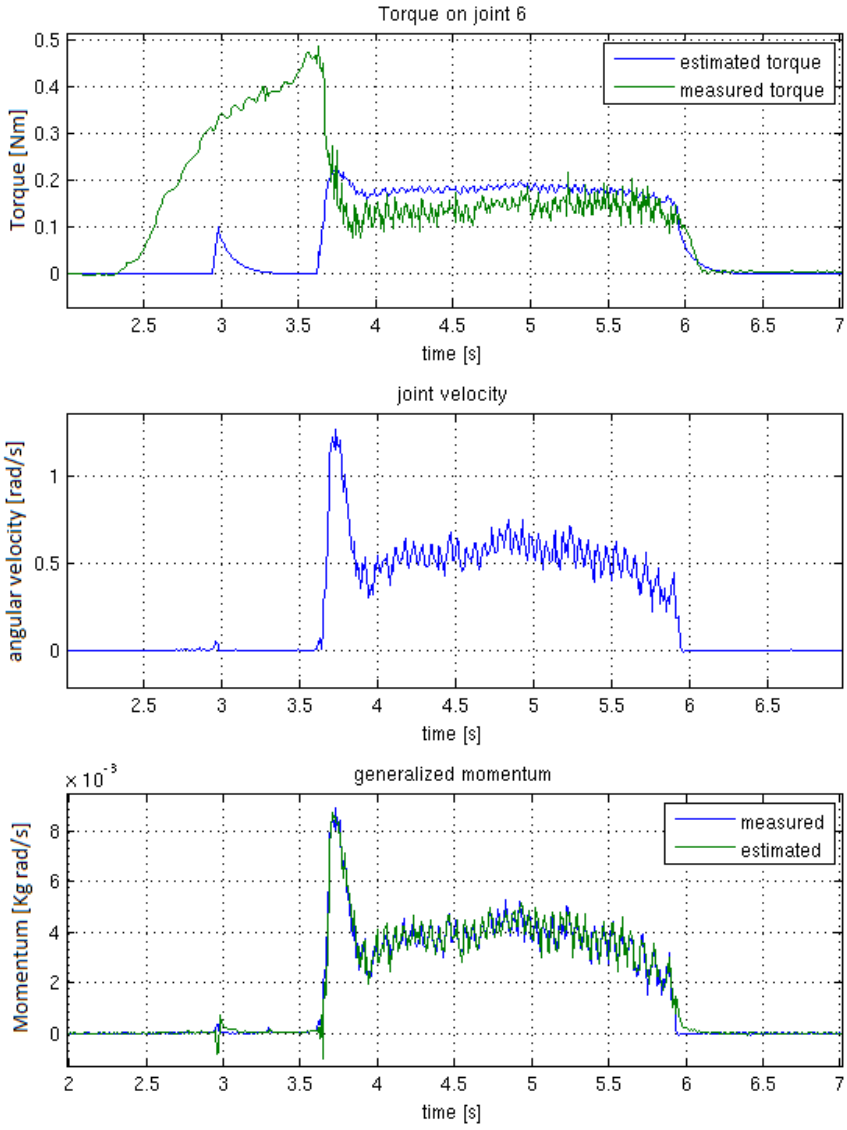


Figure 5.2: Measured and estimated torque on a joint. When the joint is not moving the observer cannot estimate the force; therefore only forces that initially overcome the static friction can be detected.

An effective strategy to reduce the required forces to overcome the static friction is to use a *dithering torque*. This method is presented and tested in the next section.

5.3 Dithering Signal to reduce Stiction

As already explained, the Kalman filter algorithm does not allow to estimate external forces when the robot is not moving, i.e. when the joint angular velocity is zero. Therefore, when the robot is not moving, just the external torques that initially overcome the joint static friction can be observed.

In order to minimize the initial external torque to start moving a joint, a strategy based on a feedforward dithering torque at almost-zero velocities is here proposed. A similar approach has been presented in [26]. Since the observer cannot detect external torques unless the joint is not moving, the use of the dithering torque will decrease the threshold of torques detection, improving the observer performance.

A high frequency signal with a square wave shape is added to the feedforward torque when the joint is not moving. The amplitude of the dithering torque is small enough to avoid the joint motion, but at the same time the resulting vibrating effect dramatically reduces the static friction in the gear box of the joint. The dithering torque is then deactivated as soon as the joint starts rotating.

Also a sinusoidal trend of the dithering signal was investigated. The sine wave was less effective than the square wave, because the beneficial affect of the always-vibrating gear box was given exactly by the high discontinuity signal of the motor torque.

The amplitude of the dithering torque was chosen equal to 30% of the Coulomb friction, since there is a trade-off between the beneficial effect of the dithering and the unpleasant vibrations felt by the user during the lead-through programming. With this magnitude of the dithering torque, the resulting amplitude of the vibration in the joint is equal to 1/100 rad (motor side) and it is assumed as an acceptable level of vibrations for the user.

A direct consequence of the dithering torque is a very noisy angular velocity measure when the joint is not moving. An experimental test was performed in order to find out the relation between the dithering frequency and the amplitude of noise in the velocity measurements. While the robot was not moving, a feedforward dithering torque with constant amplitude and increasing frequency was applied. The robot position was maintained still by feedforward gravity compensation. The resulting velocity measurements were the output of the test. In Figure 5.3, the outcome of the test is illustrated: the resulting velocity amplitude remarkably increases for frequencies higher than 15 Hz, probably due to some resonances of the system.

As a consequence, the frequency of the dithering signal was chosen equal to 15 Hz.

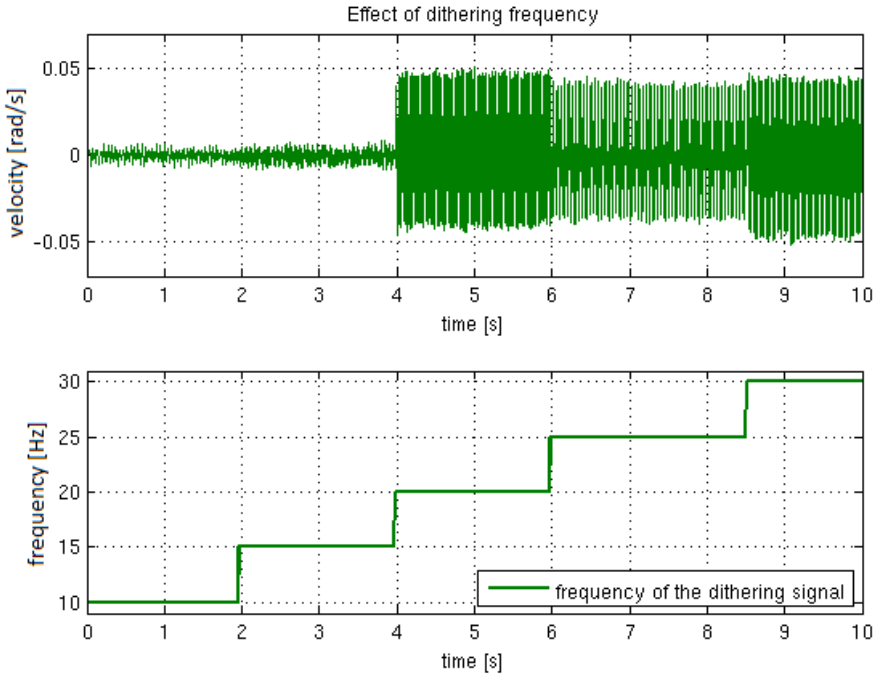


Figure 5.3: Velocity measurement noise dependency on the dithering frequency. In the test, the dithering frequency were increased keeping the amplitude constant, in order to analyze the effect on the velocity measurement.

In order to better investigate the relation between dithering frequency and the resulting vibration of the end-effector, another experiment was performed. Similarly to the previous test, the frequency of the dithering signal was increased from 0 Hz to 20 Hz, keeping the dithering amplitude constant. In order to measure the vibration level of the end-effector, the force/torque sensor presented in Section 3.1.3 was used. Since the sensor is mounted on the end-effector, a 6 components vector (3 forces and 3 torques) is measured. From the force/torque measurements, the 6 accelerations of the tool with respect to the end-effector were calculated dividing the measured force and torque by the tool mass:

$$a_i = F_i/m_{tool} \qquad i = 1,2,3 \tag{5.19}$$

$$a_i = F_i/(m_{tool}r_{tool}) \qquad i = 4,5,6 \tag{5.20}$$

where F_i is the i-measurement (the force sensor measures 3 axial components and 3 torques), m_{tool} is the mass of the tool and r_{tool} is the distance

between the sensor frame and the tool center point. Once the acceleration of the tool center of mass was measured, the 6-norm was calculated, in order to have a single parameter as an information of the amplitude of vibration of the end-effector:

$$||a||_6 = (\sum_{i=1}^{6} a_i^6)^{1/6} \tag{5.21}$$

The 6-norm of the measured acceleration of the tool is the outcome of the experiment.

In this test, the frequency was increased by 1 Hz every 2 seconds, from 0 to 20 Hz. The acceleration norm was calculated at every iteration of the real-time controller. The result of the test is illustrated in Figure 5.4. In the graph, it can be noticed that the acceleration amplitude, i.e. the vibration of the end-effector, is almost linear with the dithering frequency, with two picks at around $10Hz$ and $17Hz$ where the amplitude of vibration is larger. It is important to notice that this test has to be considered as a pure qualitative experiment, since the resonances in the vibration of the end-effector are configuration-dependent, therefore different acceleration picks could be obtained, depending on the position of the manipulator.

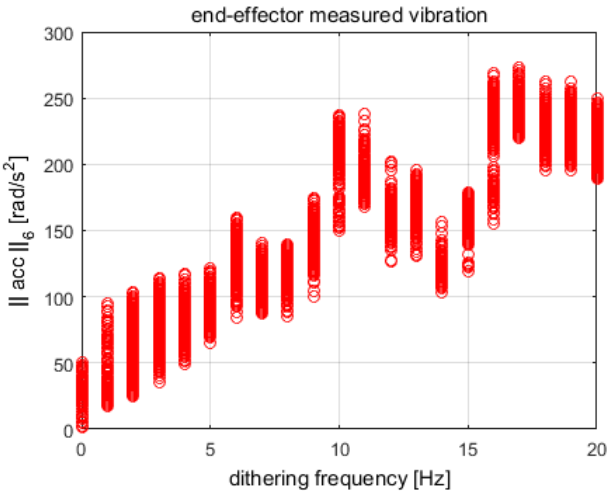


Figure 5.4: End-effector vibration dependency on the dithering frequency. The end-effector vibration, considered as the 6-norm of the measured acceleration components, depends on the dithering frequency. It shows an almost-linear trend, with two position-dependent resonances.

5.3.1 Modification of the Algorithm

The dithering torque at zero velocity represents a high-level disturbance for the Kalman filter, since it makes the velocity measurements very noisy. Although the dithering torque should be a known signal, and therefore it should be a compensable disturbance, the real motor torque is slightly different from the desired reference torque. Moreover, there are small delays between inputs and outputs of the external research interface. For these reasons, the noise in the velocity measurements due to the dithering torque cannot be compensated.

Since the velocity noise at almost-zero velocity is increased, the feedforward friction compensation torque would be very noisy too, because the velocity is the input for the computation of the feedforward torque. In order to avoid vibrations in friction compensation, the friction compensation illustrated in Section 4.2.2 was modified, enlarging the range around zero velocity in which friction is not compensated. Furthermore, the dithering feedforward torque was excluded from the observer inputs. In fact, it was subtracted from the motor torque included in the observer input defined in Equation (5.10). The presented adjustments of the algorithm were necessary to have a smoother force estimation at zero velocity.

A simulation to observe the estimation improvement was performed. In Figure 5.5, the comparison between the estimates obtained before and after the presented modifications is shown. It can be observed that the new estimate presents less noise when the joint is not rotating and the transients are smoother.

5.3.2 Torque Detection Improvement

The advantage deriving from the application of the dithering torque at almost-zero velocities is significant. In fact, the experiments proved that the external force required to start moving the robot is remarkably smaller than before.

In the tests, gravity and friction were compensated as described in Section 4.2, adding the feedforward dithering signal at almost-zero velocities. The estimated external torque is the outcome of the experiments, in which the torque measured with the sensor was used just for validation.

Figure 5.6 shows a test in which a joint of the robot is moved by hand, firstly without the dithering torque at zero velocity and secondly when the dithering is active. In the upper graph, the estimated torque is compared with the one measured with the force sensor. Looking at the measured external torque, it can be noticed how the required torque to overcome

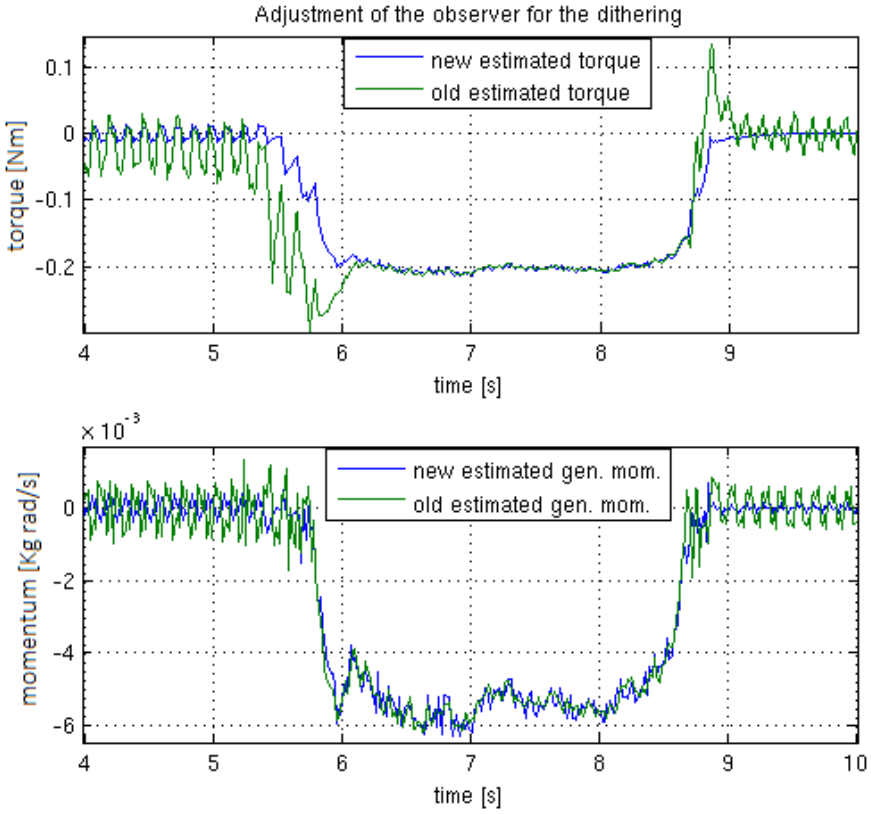


Figure 5.5: Simulation that compares the torque estimate before and after the modifications of the observer algorithm. The effect of the dithering torque at zero velocity can be compensated, in fact, the new estimation presents less noise and a smoother behavior.

the static friction is almost half of the one without dithering, resulting in a smaller initial error in force estimation. As a consequence, also the initial step in velocity is smaller. The dithering torque is automatically deactivated as soon as the joint starts rotating.

Another strategy to reduce the low-velocity friction uncertainties could be based on the use of a large integral gain of the ABB control at low velocities, as described in [25] and [24].

5.4 Discussion

In this chapter, a disturbance observer for external torque estimation has been presented. The main advantage of the generalized momentum for-

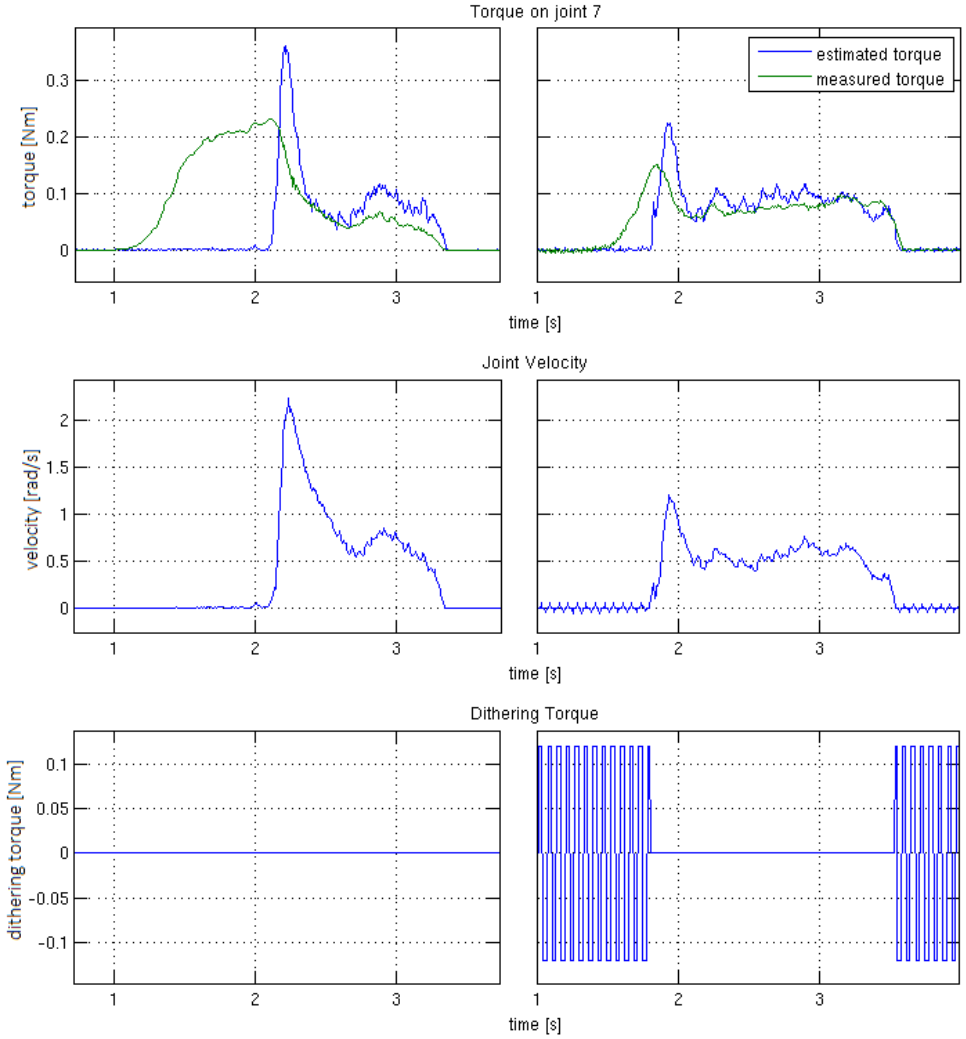


Figure 5.6: Force estimation improvement with the dithering torque. The joint is moved firstly without the dithering torque at zero velocity, secondly after the dithering has been activated. It can be noticed that the measured initial torque required to start moving the joint is significantly smaller than the first case.

mulation is that it does not require neither the inversion of the inertia matrix nor the computation of the numerical acceleration, that are generally prone to numerical issues.

Since no measurements of external torque are available, the estimate of the external torques using the Kalman filter is based on the dynamic model of the manipulator. This means that the quality of the estimated torque depends on the model of the robot, and in particular on the friction

model since it has the major uncertainties.

As illustrated in Section 4.1.3, the measured Coulomb friction torque results different depending on the angular position of the joint, making the friction values dependent on the angular position. This characteristic is possibly due to the twisting of cables inside the links and has not been modelled into the friction model. The definition of a reliable friction model is the main obstacle in the presented approach, because, since the magnitude of the estimated torque entirely depends on the friction model, the resulting position-dependent error establishes a practical limit in the performance of the admittance control. Therefore, an accurate robot modeling and force observer tuning are crucial for a good estimation of external torques.

The external torques can be estimated as soon as the joint starts rotating, because otherwise all the inputs of the observer would be null or constant. This means that only the external torques initially higher than the joint static friction can be detected. In order to mitigate this undesired characteristic of the algorithm, a strategy based on a dithering signal to reduce the static friction has been adopted. The dithering torque is applied in feedforward when the joint is not moving. It cannot actually rotate the joint, but it is sufficient to keep the gear box constantly moving, drastically reducing the external torque required to start the movement of the robot. This method turned out to be remarkably effective, in fact, the required torque to overcome the static friction is almost half of the one without dithering, leading to a smaller step in the joint velocity and lower latency between the application of the external torque and its detection.

Chapter 6

Admittance Control

In this chapter, the force control architecture is presented. It is designed to make the LTP active, e.g. the robot behavior can be controlled during the manual handling, depending on the specific task that has to be accomplished. The input of the force control are the estimates of external torques, and it provides an active control of the manipulator in parallel with the passive feedforward compensation.

6.1 Introduction

The control of the contact forces between robot and external environment permits to execute complex tasks safely, allowing a safe interaction with other physical objects or humans. Successful execution of a task that includes interactions with a stiff environment via motion control would be possible only if its geometry was known with accuracy higher than the tolerance required in the manufacturing process. This is generally infeasible, since the geometric environment is difficult to model with high accuracy. In the case of lead-through programming, if the forces exchanged between manipulator and user are not known, just a passive feedforward control can be implemented.

A feedback control can be implemented by knowing the external torques applied by the user during the handling. The advantages of an active control during the lead-through programming are obviously multiple; for instance, it is be possible to change the behavior of the robot in terms of inertia and damping, or it could be required the definition of virtual constraints to limit the work-space of the robot.

Interaction control strategies can be grouped in two categories: those performing indirect force control and those performing direct force con-

trol. The main difference between the two categories is that the former achieve force control via motion control, without explicit closure of a force feedback loop; the latter, instead, offers the possibility of controlling the contact force to a desired value, thanks to the closure of a force feedback loop. The admittance control scheme that will be here illustrated belongs to the first category.

6.2 Control Architecture

The control scheme here presented is based on the external torques estimated by the Kalman filter described in Chapter 5. An active force control permits to modify the behavior of the real robot imposing velocities and displacements of a virtual mechanical system taken as model. Since the external torques represent the input and the desired angular accelerations of joints are the output, the following control architecture is called *admittance control* ([22], Section "Impedance Control").

The equation of motion of the robot in joint-space can be written as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_{fric}(q,\dot{q}) = \tau - \tau_e \tag{6.1}$$

where M is the inertia matrix, C the Coriolis matrix, G is the gravity load on joints, F the friction component, τ is the motor torque on the arm side of the gear box and τ_e are the external torques.

In the active LTP the motor torque is calculated as the superposition of a feedforward component (gravity and a friction compensation) and a feedback component (admittance control):

$$\tau = \tau_{ff} + \tau_{fb} \tag{6.2}$$

$$\tau_{ff} = G(q) + \epsilon\tau_{fric}(q,\dot{q}) \tag{6.3}$$

$$\tau_{fb} = K_P(q_{ref} - q) + K_D(\dot{q}_{ref} - \dot{q}) \tag{6.4}$$

The feedforward component is denoted by τ_{ff}; the term ϵ is an empirical coefficient for a percentage of the modelled friction $\tau_{fric}(q,\dot{q})$ to be compensated. The feedback component τ_{fb} is the output of a PD controller. The position and velocity references are generated by the admittance control algorithm, based on a simplified model of the manipulator that includes the masses of links and an additional damping matrix:

$$\hat{\tau}_{ext} = \tilde{M}(q)\ddot{q} + \tilde{D}\dot{q} \tag{6.5}$$

Matrices $\tilde{M}$ and $\tilde{D}$ represent the inertia and the damping matrix respectively. Intuitively, the admittance control aims to make the real robot

behave like a different mechanical system; in other words, the controller guarantees that the robot acts in a certain way as response to the external inputs (forces). The robot behavior is practically determined by the admittance equation (6.5). In the experiments, matrix $\tilde{M}$ was set proportional to $M(q)$ in (5.1), while matrix $\tilde{D}$ was chose as diagonal. The resulting acceleration reference can be easily calculated as

$$\ddot{q}_{ref} = \tilde{M}(q)^{-1}(\hat{\tau}_{ext} - \tilde{D}\dot{q}), \tag{6.6}$$

Accordingly, the position and velocity references can be obtained from numerical integration of $\ddot{q}_{ref}$. The equivalent block scheme is illustrated in Figure 6.1.

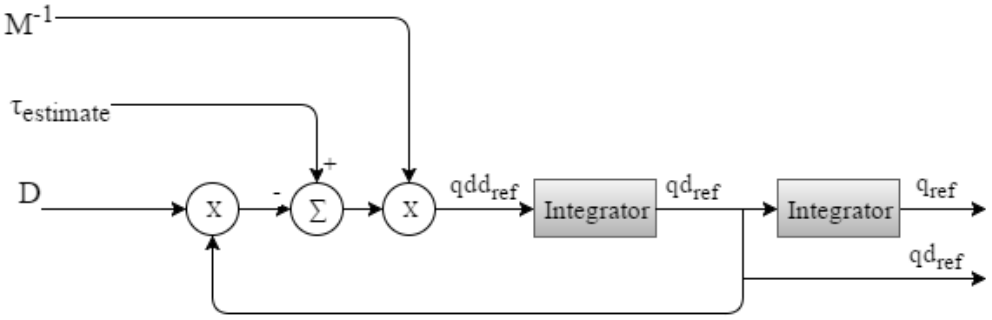


Figure 6.1: Block scheme of the simplified model to calculate the references for the admittance control.

To modify the desired behavior of the manipulator, matrices $\tilde{M}$ and $\tilde{D}$ can be tuned; for instance, if the arm is expected to react as a heavily damped system, a higher value for the $\tilde{D}$ matrix is chosen.

In order to achieve the desired position and velocity, a PD controller was designed. It would have been possible to use the internal ABB control architecture, since the external research interface [8] permits to modify the proportional, derivative and integral gains of the original controller.

The resulting block scheme of the control architecture is illustrated in Figure 6.2. As already said, the estimated external torque is the input of the admittance control, that generates the position and velocity references for the PD controller. The resulting feedback control torque is then added to the feedforward compensation torque.

6.3 Tests

All the experiments were performed on the ABB robot YuMi, using the experimental setup described in Section 3.1.

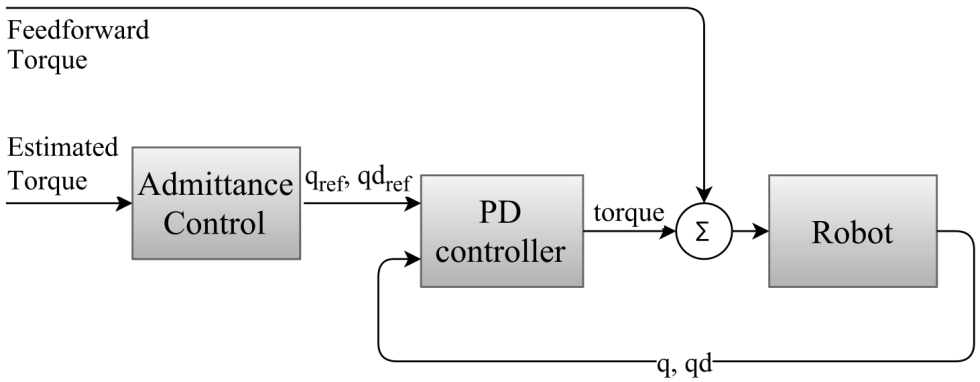
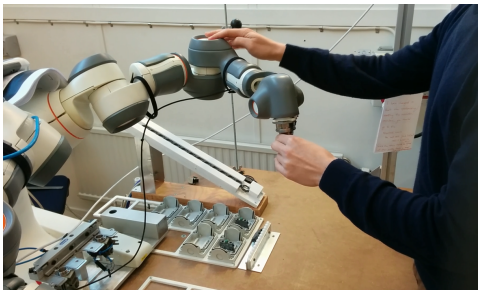


Figure 6.2: Control scheme: the estimated external torque is the input of the admittance control, which generates the references for the PD controller. The resulting motor torque is then added to the feedforward component.

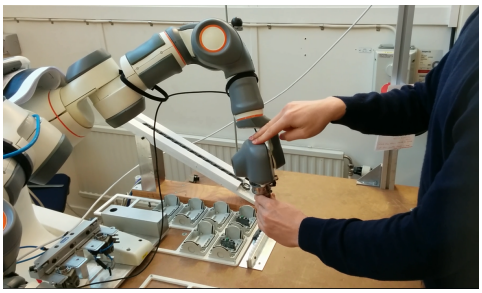
Several experiments were carried out to explore the combination of feedforward and admittance control. A first intuitive remark can be done: when the joints start moving, the feedforward compensation of friction is applied faster than the admittance control torque, because of the time needed by the observer to converge to the external force.

The active LTP was tested manually on the robot YuMi to evaluate the ease of handling in terms of required force and accuracy during small displacements. The robot appears very easy to move thanks to the feedforward compensation and to the active motor torque. In fact, just a small external force is required to move the joints, even in the case multiple joints are rotated simultaneously. The stability of the control system was inquired; the robot presents a stable behavior during LTP, both in free-space motion, i.e. the robot is not in contact with fixed objects or surfaces, and during constraint operations, i.e. when the tool gets in contact with stiff surfaces or grasps a fixed object in an assembly task. In Figure 6.3, a sequence of pictures that illustrate a teaching task is shown. In the first pictures, the robot is moved in free-space, and it is clear how moving different joints of the robot with two hands is particularly helpful to handle a redundant manipulator. Moreover, an example of LTP to teach an assembly task is illustrated; the robot has to accomplish a series of operations to assemble a red button in its case. In the latter task, the stability of the end-effector has to be guaranteed, so that the tool can be constrained during the assembly task without showing any vibration. During these tests, no stability problems were detected, and the robot resulted very easy to handle.

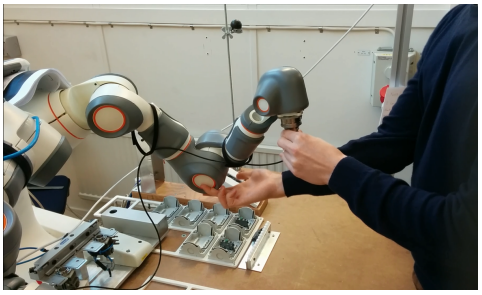
In the next sections, different experiments are described in details.



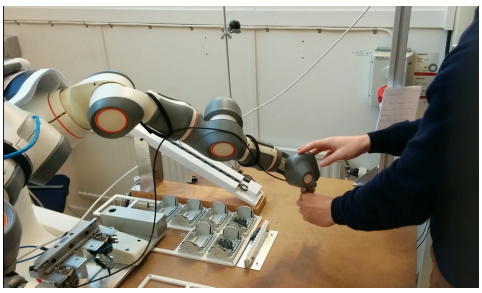
(a) Free-space motion



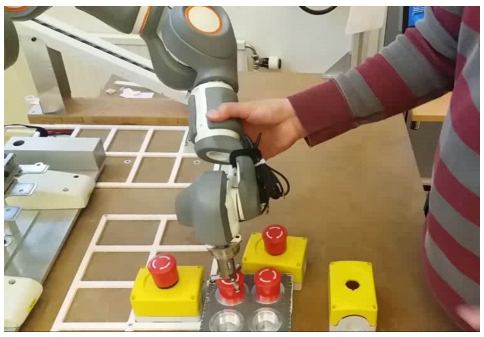
(b) Free-space motion



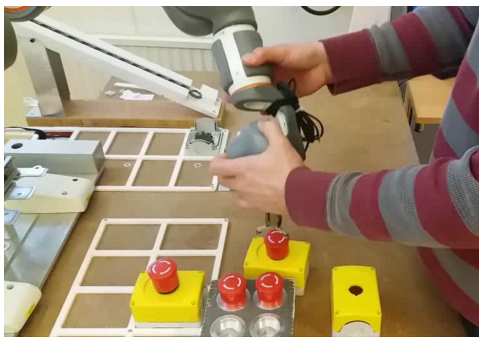
(c) Free-space motion



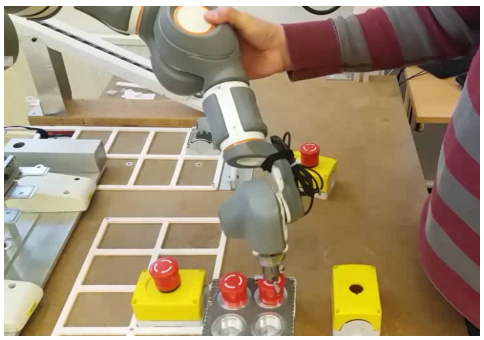
(d) Free-space motion



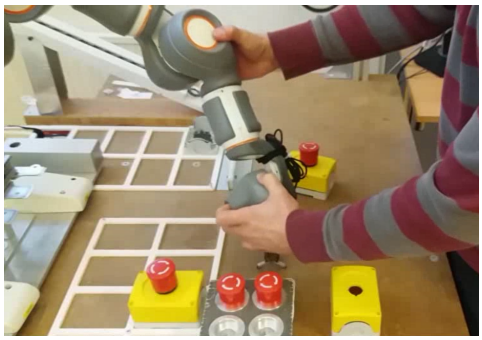
(e) Tool constrained



(f) Precision motion



(g) Tool constrained



(h) Contact with stiff surface

Figure 6.3: Teaching by demonstration: these frames illustrate free-space motion and assembly tasks during LTP. In particular, it is shown when the tool gets in contact with a stiff surface or is constrained during the assembly task.

Firstly, some examples of the admittance control response for different choices of the damping matrix $\tilde{D}$ are presented. An experiment that tests the stability limits of the active lead-through programming is illustrated; in particular, an undesired behavior of the control system due to unmodelled disturbances is shown. Finally, a comparison between the active LTP described in this thesis and a passive LTP previously implemented on the same robot is presented.

6.3.1 Damping Control

In the case that only the damping of the robot is desired to be changed, the control architecture is called *damping control* [18]. In fact, the feedback control torque of Equation (6.4) is calculated using just the term proportional to the velocity reference, i.e. setting the proportional gain of the PD controller equal to zero.

High Damping Matrix

Recalling the admittance control equation (6.6), in this experiment, the inertia matrix $\tilde{M}$ was set equal to the inertia matrix of the dynamic model of the robot, while a damping matrix $\tilde{D}$ with high values was chosen. In Figure 6.4 the result of the experiment is shown.

It can be immediately noticed that the desired velocity is lower than the measured one; as a result, the admittance control torque (red line) has opposite sign with respect to the external torque. This means that the motor torque tends to slow down the rotation and the user can feel that the robot is harder to move.

In the upper graph, the external torque measured with the force sensor (green line) is shown to demonstrate the accuracy of the estimation, even though the torque magnitude is very small, around 0.1 Nm.

Low Damping Matrix

Recalling the admittance control based on Equation (6.6), in order to facilitate the handling of the manipulator a damping matrix $\tilde{D}$ with small values has to be chosen. In fact, setting a low damping matrix will generate a desired velocity higher than the measured one, since it models the behavior of a low-damped system. Therefore, the active LTP is supposed to actively support the movement of the joints.

An experiment in which an external torque was applied is illustrated in Figure 6.5. It can be seen that the admittance control torque (red line)

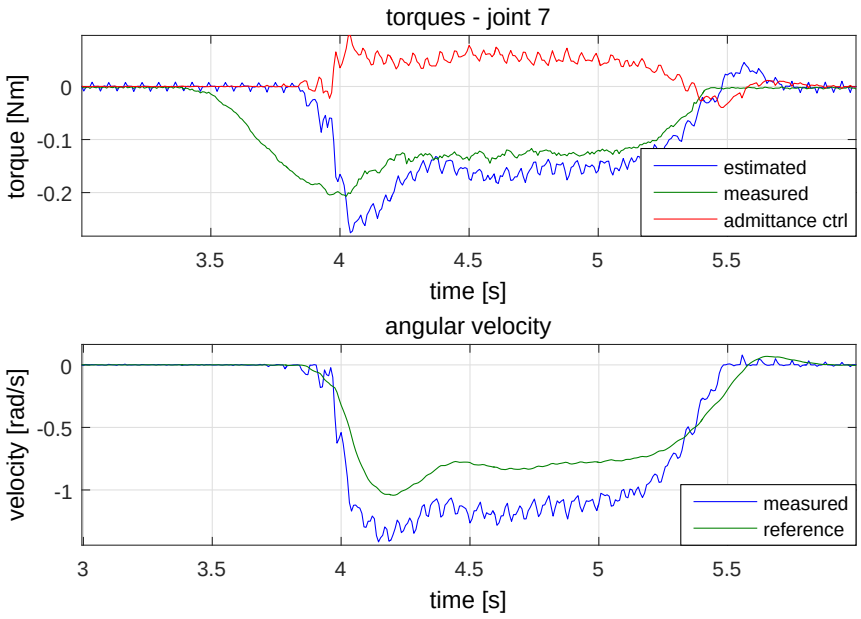


Figure 6.4: Admittance control: high damping. In this experiment a damping matrix $\tilde{D}$ with high values was set. It can be noticed that the resulting desired velocity is lower than the real one, therefore the control acts a damping effect on the system.

compensates for the overshoot and undershoot of the force estimation at the beginning and at the end of the movement, and has the same sign as the external torque in the middle. Thus, the motor actively favors the rotation of the joint. In the upper graph, the external torque measured with the force sensor (green line) is shown to demonstrate the accuracy of the estimation. Since this experiment concerns the last joint of the robot (where the tool is mounted), the order of magnitude of the torques is extremely small (around 0.1 Nm), and for this reason the uncertainties in the friction model and the consequent error in torque estimation are emphasized.

6.3.2 Instability for Low Damping and Joint-End

In the experiment presented in this section, the active LTP was studied in critical conditions. In fact, a very low damping matrix $\tilde{D}$ was set; furthermore, high proportional and derivative gains of the PD controller presented in Equation (6.4) were chosen.

In Figure 6.6 the results of the test are shown. The real external

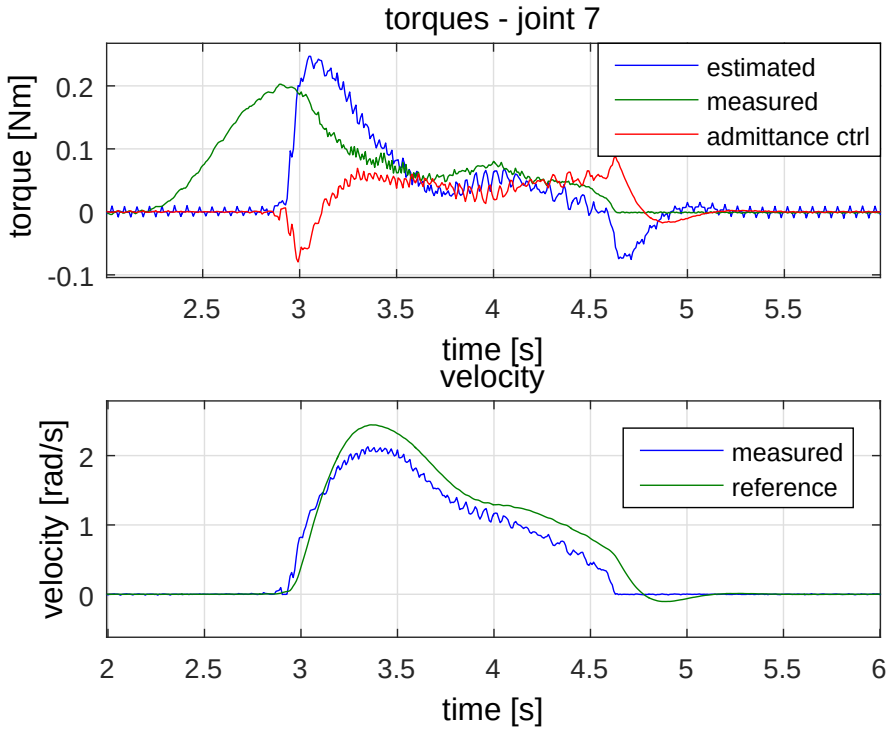


Figure 6.5: Admittance control: low damping. In this experiment a damping matrix $\tilde{D}$ with low values was set. It can be noticed that the admittance control compensates for the overshoot and undershoot of the force estimation at the beginning and at the end of the movement, respectively. The calculated desired velocity is slightly higher than the measured one.

force was applied just at the beginning, as measured by the force sensor (green line), which was used just for validation. The torque estimation is not very accurate, because of the position-dependent uncertainty in the friction model (the magnitude of the external torque is very low since this test concerns the last joint of the arm). At $3.6s$ an abrupt unmodelled disturbance appears: the joint reaches its rotation end, and this event is interpreted by the observer as an external torque that aims to rotate the joint in the opposite direction. Consequently, the low-damped admittance control reacts to this estimated torque with a motor torque in the same direction, causing an undesired rotation of the joint.

This undesired behavior of the control system can be avoided reducing the gains of the PD controller, in particular the proportional gain, and by choosing a damping matrix with sufficiently large values.

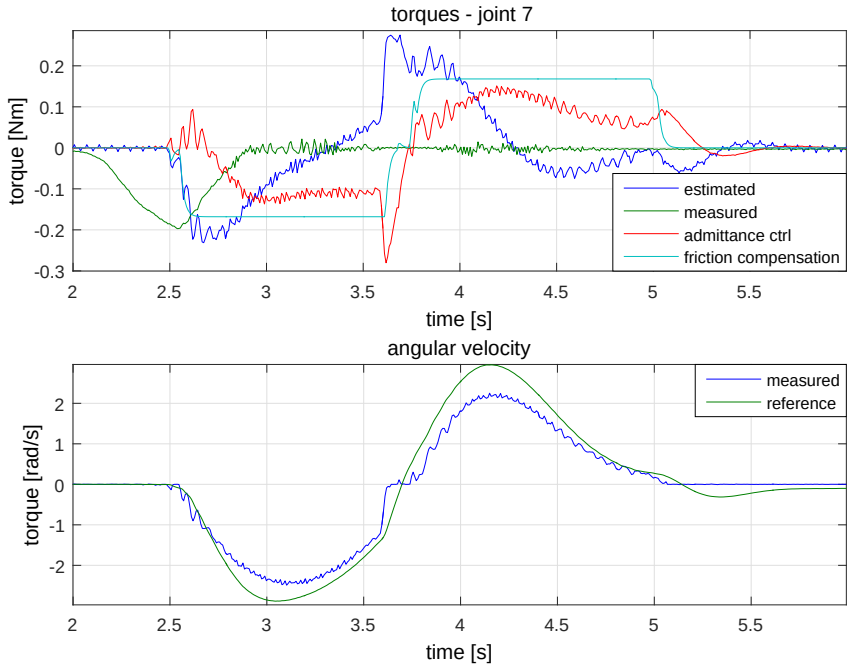


Figure 6.6: Instability experiment: test with small damping matrix and high PD control gains. The external torque is applied just in the initial phase (green line). At $3.6s$ the joint reaches the rotation end; this unmodelled event is seen by the disturbance observer as an external torque. Due to delays, uncertainties in friction model and high gains of the PD controller, the resulting motor torque (light blue + red lines) makes the joint rotating in the opposite direction.

6.3.3 Comparison with a Passive LTP

The active lead-through programming implemented in this thesis was compared to a passive LTP, in order to obtain a qualitative idea of the improvements achieved with the active control. The passive LTP implemented in the Robotics Lab of Lund University was used as the benchmark. More details can be found in [25], [24] and [14].

For a rigorous comparison between the two different implementations, it would have been necessary to adopt a second robot, in order to accomplish exactly the same displacements and velocities while moving YuMi's arm and measuring the applied external forces, in different experiments. This was unfortunately impossible because of the lack of a second manipulator, in fact, during the development of this thesis the second arm of YuMi was out of order.

In this test, a joint of YuMi was rotated manually and the applied force

was measured by the force sensor mounted on the end-effector. The test was repeated twice, once using the active LTP and the second time using the passive implementation. The data of the two different experiments have been overlapped in Figure 6.7. Since the external torque was applied by hand, the resulting displacement and velocity are not identical in these two experiments. However, as it can be seen in the lower graph in Figure 6.7, the resulting velocities are very similar, hence a similar value of joint friction during the rotation can be assumed. In this comparison, the required external torque for rotating the joint in the passive LTP is almost twice as much as in the active implementation. This result demonstrates that the active LTP can be successively employed in order to facilitate the trajectory teaching. It can also be noticed that, despite a lower level of external forces, in the active implementation the onset of motion is earlier.

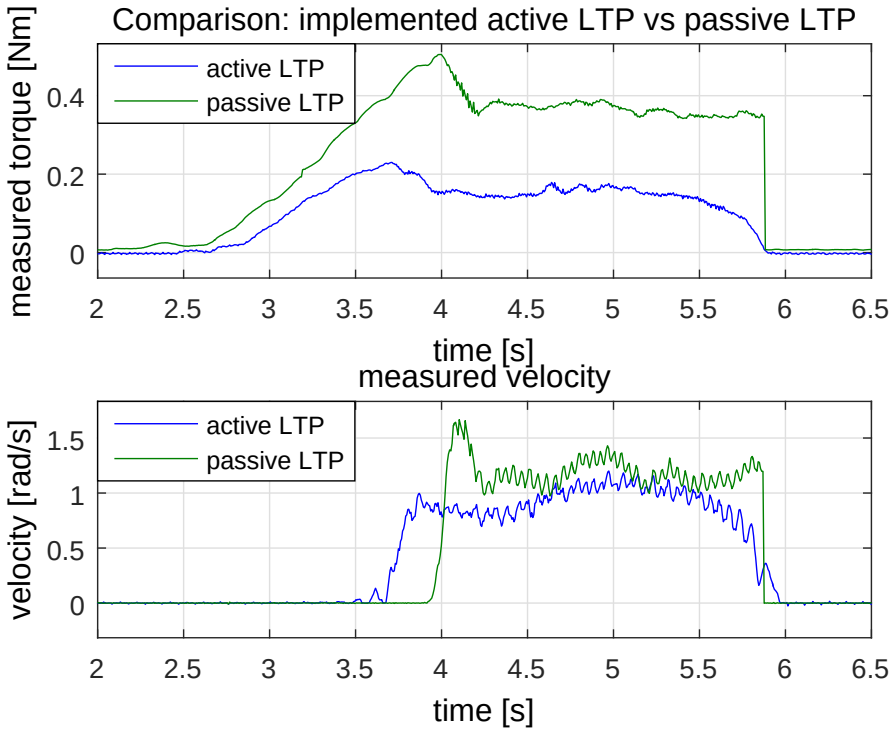


Figure 6.7: Comparison between a passive lead-through programming previously implemented on YuMi [25] and the active LTP presented in this thesis. Using the two different implementations, the external forces (measured with a force sensor) were compared for generating a similar movement of a joint. It can be noticed that, with the active LTP, the required external torque is almost half of the passive solution, and the joint starts rotating earlier.

6.4 Discussion

In this section, the admittance control architecture that allows to make the LTP active has been presented.

A hybrid control architecture has been implemented, in fact, the feedback control torque is added to the feedforward model-based compensation of gravity load and joint friction. The combination of admittance control and feedforward friction compensation has proven useful specially when the joints start moving. In fact, the feedforward torque guarantees an immediate start of the movement. Therefore, it reduces the non-linearity of the system and contributes to its stability. The resulting control torque makes the robot easy to handle and smooth in the movements, showing a stable behavior also in case of contact with stiff surfaces.

The active force control has been developed on the base of the estimated external torques; it is called admittance control since it is part of the category of indirect force control, with the force as input and a motion reference as output. The resulting controller can be tuned in order to modify the robot behavior in terms of inertia and damping, so to facilitate the handling of the robotic arm. Furthermore, it potentially includes the introduction of virtual constraints during lead-through programming, in order to guarantee safe human-robot interaction or object avoidance.

The stability of the system is strictly related to the choice of the controller gains. It was shown that the assignment of proper controller parameters makes the robot behave in the desired way.

In order to outline the improvements of the active LTP with respect to a passive one, a comparison with a passive LTP previously implemented on the same robot was carried out [25]. The experiment demonstrated remarkable improvements in terms of ease of handling and flexibility. In fact, the required external force to move a joint with the active implementation is almost half the one required by the passive approach. It has been also noticed that, despite a lower level of external forces in the active implementation, the onset of motion is earlier.

Chapter 7

Conclusion and Future Work

In conclusion, the sensorless active lead-through programming here presented was implemented and tested on the robot ABB YuMi fulfilling all the expectations, and all the planned goals of this thesis project were achieved.

The complete implementation of a sensorless active lead-through programming was presented; it allows the operator to manually handle the robot in order to teach a desired task. The control system has been denoted as *active*, since it actively supports the desired operations via an admittance control architecture. In fact, the controller can be tuned ad hoc, in order to change the robot behavior in terms of inertia and damping with respect to the external forces applied by the user. The resulting control makes the robot extremely smooth to handle and stable also in critical task, e.g. when it gets in contact with stiff surfaces or becomes constrained during contact. In comparison with the passive LTP previously implemented on the same robot [25], the active LTP shows improvements in terms of ease of handling and flexibility. In fact, the required external force to move a joint with the active implementation is almost half the one required by the passive approach.

In order to implement an active LTP without using force sensors, the external torques are estimated by a Kalman filter in joint space based on the generalized momentum formulation. This approach is particularly convenient since it does not require neither the inversion of the inertia matrix nor the computation of the numerical acceleration, while estimating the external torques on all the joints of the robot. The resulting force observer is able to detect the external torques on all the joints of the robot, with a good level of accuracy.

The main limitations that were encountered during the implementation of the observer concern the torque estimation when the robot is still,

Figure 7.1: The ABB YuMi robot in the Robotics Lab used in the experimental tests.

and the accuracy of the estimated torque magnitude. The former limitation interests the detection of external torques when the robot joints are not moving. In fact, an intrinsic characteristic of the presented approach is that the external torque cannot be detected if the joint velocity is equal to zero, therefore, only the torques that initially overcome the joint static friction can be correctly estimated. To mitigate this undesired characteristic, a strategy based on a feedforward dithering signal at zero velocity was adopted. This solution was demonstrated to be effective, since it remarkably reduces the static friction and, consequently, the initial estimation error. The second limitation of the presented methodology concerns the magnitude of the estimated torques. In fact, it is based on the dynamic model of the manipulator, since no measurements of external torque are available. This means that the magnitude of the estimated torque basically depends on the the model of the robot, and in particular on the friction model, since the friction parameters have the major uncertainties. Therefore, an accurate friction model is crucial in order to obtain correct torque estimations.

A relevant future development would concern the improvement of the friction model. In this thesis a velocity-dependent friction model, described as a Gaussian variable, was developed, in which the model parameters were experimentally defined for each joint. During the friction identification experiments, a position-dependency of the friction torque was observed. In fact, the friction magnitude considerably differs for different angular positions of the same joint; this could be due to the twisting of the cables inside the robot links. A friction model that considers the position-dependency of the friction torque in addition to the velocity-dependency would lead to more accurate torque estimations. Since such a

complex model would involve a large number of empirical parameters, it would be interesting to investigate automated procedures for parameters acquisition.

Further research could be carried out on the stability of the active LTP here presented. In fact, both the tuning of the matrices of the admittance control algorithm and both the choice of the PD controller gains have a large impact on the robot behavior during lead-through programming. In particular, in the presented experiments, a derivative control was preferred to a complete PD controller, mostly because the damping reduction was the main goal.

The force control allows for modifying the behavior of the robot in multiple scenarios and makes it possible to introduce virtual constraints during lead-through programming. For instance, the handling of the manipulator could be limited to a given Cartesian plane in the work-space, in order to increase the accuracy of LTP. Another example of virtual constraint could be the 3D limitation of the work space of the robot, so that a safe human-robot interaction can be guaranteed. Furthermore, obstacle avoidance algorithms can be easily implemented using the admittance control, in order to assist the operator during kinesthetic teaching.

Active lead-through programming is an extremely relevant topic of research in the modern industry. In fact, the automation of the production processes always requires more efficient programming methods and easily reconfigurable production lines, so offering LTP countless employment possibilities.

References

[1] Abb yumi. http:// new.abb.com/ products/ robotics/ yumi. Last accessed: 2016-02.

[2] Ati force/torque transducer. http:// www.ati-ia.com/ products/ ft/ sensors.aspx. Last accessed: 2015-12.

[3] Labcomm. http:// www.control.lth.se/ Research/ Tools. Last accessed: 2015-12.

[4] Xenomai. http:// www.xenomai.org. Last accessed: 2015-12.

[5] A. Alcocer, A. Robertsson, A. Valera, and R. Johansson. Force estimation and control in robot manipulators. In J. Sasiadek and I. Duleba, editors, *Robot control 2003*, pages 31–36. Elsevier.

[6] V. Ang, L. Wei, and S. Y. Lim. An industrial application of control of dynamic behavior of robots - a walk-through programmed welding robot. *Int. Conf. Robotics and Automation (ICRA)*, 3:2352–2357, 2000.

[7] L. Bascetta, G. Ferretti, G. Magnani, and P. Rocco. Walk-through programming for robotic manipulators based on admittance control. *Robotica*, 31:1143–1153, 10 2013.

[8] A. Blomdell, G. Bolmsjö, T. Brogårdh, P. Cederberg, M. Isaksson, R. Johansson, M. Haage, K. Nilsson, M. Olsson, T. Olsson, A. Robertsson, and J. Wang. Extending an industrial robot controller–Implementation and applications of a fast open sensor interface. *IEEE Robotics & Automation Magazine*, 12(3):85–94, 9 2005.

[9] A. Blomdell, I. Dressler, K. Nilsson, and A. Robertsson. Flexible application development and high-performance motion control based on external sensing and reconfiguration of ABB industrial robot controllers. In *Proc. ICRA 2010 Workshop on Innovative Robot Control*

Architectures for Demanding (Research) Applications, pages 62–66, Anchorage, Alaska, May 2010.

[10] S. Calinon and A. Billard. Active teaching in robot programming by demonstration. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pages 702–707, Aug 2007.

[11] P. I. Corke. *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011.

[12] A. De Luca and R. Mattone. Actuator failure detection and isolation using generalized momenta. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on on Automation Science and Engineering*, volume 1, pages 634–639, September 2003.

[13] K. Eom, I. Suh, W. Chung, and S. Oh. Disturbance observer based force control of robot manipulator without force sensor. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 4, pages 3012–3017 vol.4, May 1998.

[14] M. Ghazaei. Topics in trajectory generation for robots. Licentiate Thesis 3265, Department of Automatic Control, Lund University, Sweden, Mar. 2015.

[15] G. Grunwald, G. Schreiber, A. Albu-Schaffer, and G. Hirzinger. Programming by touch: the different way of human-robot interaction. *Industrial Electronics, IEEE Transactions on*, 50(4):659–666, Aug 2003.

[16] R. Johansson, A. Robertsson, K. Nilsson, and M. Verhaegen. State-space system identification of robot manipulator dynamics. *Mechatronics*, 10(3):403–418, 2000.

[17] H. Kwakernaak and R. Sivan. *Linear optimal control systems*, volume 1. Wiley-interscience, New York, 1972.

[18] W. S. Levine. *The control handbook*. CRC press, 1996.

[19] M. Linderoth, A. Stolt, A. Robertsson, and R. Johansson. Robotic force estimation using motor torques and modeling of low velocity friction disturbances. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3550–3556, Nov 2013.

[20] Z. Pan, J. Polden, N. Larkin, S. V. Duin, and J. Norrish. Recent progress on programming methods for industrial robots. *Robotics and Computer-Integrated Manufacturing*, 28(2):87 – 94, 2012.

[21] M. Ragaglia, A. M. Zanchettin, L. Bascetta, and P. Rocco. Accurate sensorless lead-through programming for lightweight robots in structured environments. *Robotics and Computer Integrated Manufacturing*, 39:9–21, June 2016.

[22] B. Siciliano and O. Khatib. *Springer Handbook of Robotics*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[23] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[24] A. Stolt. *On Robotic Assembly using Contact Force Control and Estimation*. PhD thesis, Department of Automatic Control, Lund University, Sweden, Oct. 2015.

[25] A. Stolt, F. Bagge Carlson, M. Ghazaei, I. Lundberg, A. Robertsson, and R. Johansson. Sensorless friction-compensated passive lead-through programming for industrial robots. pages 3530–3537, Hamburg, Germany, 2015. IEEE.

[26] A. Stolt, A. Robertsson, and R. Johansson. Robotic force estimation using dithering to decrease the low velocity friction uncertainties. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3896–3902, May 2015.

[27] T. Tsumugiwa, R. Yokogawa, and K. Yoshida. Stability analysis for impedance control of robot for human-robot cooperative task system. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 3883–3888 vol.4, Sept 2004.

[28] M. Van Damme, B. Beyl, V. Vanderborght, V. Grosu, R. Van Ham, I. Vanderniepen, A. Matthys, and D. Lefeber. Estimating robot end-effector force from noisy actuator torque measurements. In *Int. Conf. Robotics and Automation (ICRA)*, pages 1108–1113, May 2011.

[29] A. Wahrburg, E. Morara, G. Cesari, B. Matthias, and H. Ding. Cartesian contact force estimation for robotic manipulators using kalman filters and the generalized momentum. Gothenburg, Sweden, 2015.

IEEE International Conference on Automation Science and Engineering.

[30] S. Wrede, C. Emmerich, R. Grünberg, A. Nordmann, A. Swadzba, and J. Steil. A user study on kinesthetic teaching of redundant robots in task and configuration space. *Journal of Human-Robot Interaction*, 2(1):56–81, 2013.

Acronyms

RNE Recursive Newton-Euler

The RNE method is a computationally efficient algorithm used for the calculation of the position-dependent inertia matrix and gravity load on the joints.

LTP Lead-Through Programming

The LTP, also called *kinesthetic teaching*, is a technique to teach the robot a trajectory by demonstration.

DH Denavit-Hartenberg

The DH notation is a possible convention for the definition of the robot kinematics.

PID Proportional-Integrative-Derivative

The PID controller is a well known control loop feedback mechanism (controller) commonly used in industrial control systems.

PD Proportional-Derivative

The PD controller is equivalent to the PID controller without the integrative part.

OLP Off-line Programming

The OLP is a modern method for robot programming, usually based on the 3D model of the complete robot work cell.

RPAR Robot Programming using Augmented Reality

DRE Differential Riccati Equation

ARE Algebraic Riccati Equation

TCP Tool Center Point

The TCP is the origin of the coordinate system of the tool.

Appendix A

RNE method

In this appendix, the Matlab code for the computation of the inertia matrix of the YuMi robot is presented. The function is based on the well known RNE algorithm, and it was adapted in order to be compiled on the real-time controller of YuMi. The inputs of the function are the joints positions, while the output is the manipulator (position-dependent) inertia matrix given in vector shape.

The RNE function that was used for the computation of the gravity load on the joints is slightly different; further details can be found in [11].

```
%%%%% COMPUTATION OF MANIPULATOR INERTIA MATRIX
%%%%% RNE METHOD - MARTINO CAPURSO - 31/10/15

function inertiaVec =RNE_M(q)

% TOOL MASS AND COG POSITION
Mtool=0.25;
Ptool=[0 0 0.04];

% transform q in a column vector
if size(q,2)>1
    q=q';
end

qd=zeros(7,1);
tau=zeros(7,1);

n=length(q);
z0 = [0;0;1];
grav=[0;0;0];
fext = zeros(6, 1);

% joint revolut offset
```

```
offset=[-pi -pi 0 -pi/2 pi pi 0]';
q=q+offset;

% Denavit-Hartenberg parameters
%      theta    d      a     alpha
dhpar=[q(1)   0.11    0.03    pi/2;
       q(2)     0     0.03    pi/2;
       q(3)  0.2465  0.0405  -pi/2;
       q(4)     0    0.0405  -pi/2;
       q(5)   0.265  0.0135  -pi/2;
       q(6)     0     0.027  -pi/2;
       q(7)   0.032    0        0];

% homogeneous transform
trans=zeros(4,4,n);
Rm=zeros(3,3,n);
pstarm=zeros(3,n);

for j=1:n
    teta=dhpar(j,1);
    d=dhpar(j,2);
    a=dhpar(j,3);
    alpha=dhpar(j,4);

    cT=cos(teta);  sT=sin(teta);
    cA=cos(alpha); sA=sin(alpha);

    Tj=[cT   -sT*cA    sT*sA   a*cT;
        sT    cT*cA   -cT*sA   a*sT;
         0      sA       cA       d;
         0       0        0       1];

    trans(:,:,j)=Tj; % homogeneous transform matrix
    Rm(:,:,j)=Tj(1:3,1:3); % rotation matrix
    pstar = [a; d*sin(alpha); d*cos(alpha)];
    pstarm(:,j) = pstar;
end

% position of links COG
rm=[-0.0103   -0.0225    0.0035;
    -0.0078    0.0141    0.0790;
    -0.0107    0.0164    0.0036;
    -0.0155    0.0180    0.0580;
    -0.0112    0.0723    0.0149;
    -0.0143    0.0060   -0.0086;
    -0.0004   -0.0003    0.0105];
rm(end,:)=Ptool;

% mass of links
```

```
mm=[1.58 1.22 1.01 0.485 0.19 0.359 0.0344]';
mm(end)=Mtool;

% inertia matrices of links
Imm=zeros(3,3,7);
Imm(:,:,1)=[0.0037    0.0007    0.0001;
            0.0007    0.0024    0.0002;
            0.0001    0.0002    0.0034];
Imm(:,:,2)=[0.0056   -0.0002    0.0008;
           -0.0002    0.0056   -0.0012;
            0.0008   -0.0012    0.0015];
Imm(:,:,3)=[0.0019    0.0005    0.0000;
            0.0005    0.0014    0.0000;
            0.0000    0.0000    0.0016];
Imm(:,:,4)=[0.0020    0.0000    0.0000;
            0.0000    0.0020   -0.0000;
            0.0000   -0.0000    0.0010];
Imm(:,:,5)=[0.4140    0.0055    0.1080;
            0.0055    0.4500    0.0308;
            0.1080    0.0308    0.0952].*1e-3;
Imm(:,:,6)=[0.2140   -0.0222   -0.0218;
           -0.0222    0.2390    0.0097;
           -0.0218    0.0097    0.2210].*1e-3;
Imm(:,:,7)=[0.6710   -0.0122    0.0059;
           -0.0122    0.6420    0.0086;
            0.0059    0.0086    0.8000].*1e-5;

% motor gear ratio
gear=[100 100 100 -100 100 -101 100];

% motors inertia
Jm_vec=[8.2 8.2 1.97 0.51 0.51 0.51 1.97]*1e-6;

%%% base frame orientation of YuMi's arms (right or left):
    % homogeneous transform of the base frame RIGHT
    %  baset=[0.5784   -0.1049    0.8090         0.0004;
    %         0.6143    0.7085   -0.3474         0.002;
    %        -0.5367    0.6979    0.4742        -0.0021;
    %            0         0         0           1];

    % homogeneous transform of the base frame LEFT
    baset=[0.5784    0.1049    0.8090         0;
           -0.6143    0.7085    0.3474         0;
           -0.5367   -0.6979    0.4742         0;
               0         0         0           1];

inertiaM=zeros(7);
for ii=1:7
    qdd=zeros(7,1);
```

```
122     qdd(ii)=1;
123
124         Fm = zeros(3,7);
125         Nm = zeros(3,7);
126
127         % rotate base velocity and accel into L1 frame
128         Rb = baset(1:3,1:3)';
129         w = Rb*zeros(3,1);
130         wd = Rb*zeros(3,1);
131         vd = Rb*grav;
132
133         %  THE FORWARD RECURSION
134         for j=1:n
135             Rt = Rm(:,:,j).';
136             I=Imm(:,:,j);
137             pstar = pstarm(:,j);
138             r = rm(j,:);
139
140                 % revolute axis
141                 wd = Rt*(wd + z0*qdd(j) + ...
142                     cross(w,z0*qd(j)));
143                 w = Rt*(w + z0*qd(j));
144                 vd = cross(wd,pstar) + ...
145                     cross(w, cross(w,pstar)) +Rt*vd;
146
147                 %whos
148                 vhat = cross(wd,r.') + ...
149                 cross(w,cross(w,r.')) + vd;
150                 F = mm(j)*vhat;
151                 N = I*wd + cross(w,I*w);
152                 Fm(:,j) = F;
153                 Nm(:,j) = N;
154         end
155
156         % THE BACKWARD RECURSION
157         f = fext(1:3); % force/moments on end of arm
158         nn = fext(4:6);
159
160         for j=n:-1:1
161             pstar = pstarm(:,j);
162             G=gear(j);
163             Jm=Jm_vec(j);
164
165             if j == n
166                 R = eye(3,3);
167             else
168                 R = Rm(:,:,j+1);
169             end
170
```

```
171             r = rm(j,:);
172             nn = R*(nn + cross(R.'*pstar,f)) + ...
173                 cross(pstar+r.',Fm(:,j)) + ...
174                 Nm(:,j);
175             f = R*f + Fm(:,j);
176             R = Rm(:,:,j);
177 
178                 % revolute joint
179                 t = nn.'*(R.'*z0)+ ...
180                     G^2 * Jm*qdd(j) ;
181 
182                 tau(j) = t;
183         end
184 
185     inertiaM(ii,:)=tau;
186 end
187 
188 inertiaVec=reshape(inertiaM,[49,1]);
```

Appendix B

Dynamic model of YuMi's arm

In this appendix, the Matlab code for the definition of the dynamic model of YuMi is presented. The Matlab code employs functions from Corke's Robotic Toolbox [11]. The dynamic model of the manipulator was used in the simulation phase in order to test the algorithms before the implementation on the real robot. The geometrical and physical data were provided by data-sheets, while the friction parameters were defined via identification experiments.

```
%%% DYNAMIC MODEL OF ABB YuMi
%%% MARTINO CAPURSO - December 2015
% (ref.: Peter Corke - Robotics, vision and control)

% definitioin Denavit-Hartenberg parameters
            Lz0=0.049;
Lx1=-0.03;  Lz1=0.11;
Lx2=0.03;   Lz2=0.1465;
Lx3=0.0405; Lz3=0.1;
Lx4=0.1395; Lz4=0.0405;
Lx5=0.1255; Lz5=-0.0135;
Lx6=0.032;  Lz6=0.027;
Lx7=0;      Lz7=0;

% definition of links
L1=Link([pi Lz1  -Lx1   pi/2]);
L2=Link([0 0  Lx2 pi/2]);
L3=Link([0 Lz2+Lz3 Lx3 -pi/2]);
L4=Link([0 0 Lz4  -pi/2]);
L5=Link([0 Lx4+Lx5   abs(Lz5)  -pi/2]);
L6=Link([0 0 Lz6   -pi/2]);
L7=Link([0 Lx6 0 0]);
L=[L1 L2 L3 L4 L5 L6 L7];
arm= SerialLink(L, 'name', 'arm');
```

```
25
26 % base offset
27 rot2 = rotx(-0.6322)*roty(0.9424)*rotz(-0.1793); %[rad]
28 t2 = 1e-6*[42.256 62.566 404.92]'; %[m]
29 arm.base = [rot2 t2; 0 0 0 1];
30
31 % link offsets
32 offset=[-pi -pi 0 -pi/2 pi pi 0]';
33 arm.links(1).offset = offset(1);
34 arm.links(2).offset = offset(2);
35 arm.links(3).offset = offset(3);
36 arm.links(4).offset = offset(4);
37 arm.links(5).offset = offset(5);
38 arm.links(6).offset = offset(6);
39 arm.links(7).offset = offset(7);
40
41 %% dynamic parameters
42
43 % motor inertia
44 Jm=[8.2 8.2 1.97 0.51 0.51 0.51 1.97]*1e-6;
45
46 % motor gear ratios
47 gear=[100 100 100 -100 100 -101 100];
48
49 % saturation torques
50 tau_max=[30 30 30 20 20 10 10]'; % (hypothesis)
51
52 % friction parameters (experimentally defined)
53 fric=[ 0.5 0.7    1  0.8  0.2  0.2 0.05]./(abs(gear).^2);
54 coul=[1.65 1.45 0.85 0.75 0.25 0.4 0.22]./abs(gear);
55
56 % link 1
57 arm.links(1).m=1.58; % mass
58 r1=[-0.0197 0.00352 0.0875]; % Cartesian coord. of COM 1 wrt frame 0
59 G10=transl(r1); % homogeneous transform of xyz
60 T10=arm.links(1).A(qnom(1));
61 G11=inv(T10)*G10;
62 arm.links(1).r=transl(G11)'; % COG 1 is defined wrt frame 1
63 Ixx=0.00373; Iyy=0.00343; Izz=0.00244;
64 Ixy=-0.000125; Ixz=-0.000719; Iyz=0.000203;
65 I10=[Ixx Ixy Ixz;
66      Ixy Iyy Iyz;
67      Ixz Iyz Izz];
68 R10=rotz(pi)*rotx(pi/2); % rot. matrix from frame 0 to frame 1
69 arm.links(1).I=R10*I10*R10';
70 arm.links(1).B=fric(1);
71 arm.links(1).Tc=coul(1);
72 arm.links(1).Jm=Jm(1);
73 arm.links(1).G=gear(1);
```

```
74
75  % link 2
76  arm.links(2).m=1.22;
77  r2=[0.0222 0.0141 0.079];
78  G21=(R10'*r2')';
79  T21=arm.links(2).A(qnom(2));
80  G22=inv(T21)*transl(G21);
81  arm.links(2).r=transl(G22)';
82  Ixx=0.00559; Iyy=0.00559; Izz=0.00146;
83  Ixy=-0.000249; Ixz=0.000751; Iyz=-0.0012;
84  I2=[Ixx Ixy Ixz;
85      Ixy Iyy Iyz;
86      Ixz Iyz Izz];
87  arm.links(2).I=I2;
88  arm.links(2).B=fric(2);
89  arm.links(2).Tc=coul(2);
90  arm.links(2).Jm=Jm(2);
91  arm.links(2).G=gear(2);
92
93  % link 3
94  arm.links(3).m=1.01;
95  r3=[0.0298 0.00358 0.0836];
96  T2=[0 0 Lz2]; % frame 2 is just translated
97  G32=r3+T2;
98  T32=arm.links(3).A(qnom(3));
99  G33=inv(T32)*transl(G32);
100 arm.links(3).r=transl(G33)';
101 Ixx=0.00193; Iyy=0.00162; Izz=0.00144;
102 Ixy=-3.83e-5; Ixz=0.000466; Iyz=-1.23e-5;
103 I30=[Ixx Ixy Ixz;
104       Ixy Iyy Iyz;
105       Ixz Iyz Izz];
106 R30=rotx(-pi/2);
107 arm.links(3).I=R30*I30*R30';
108 arm.links(3).B=fric(3);
109 arm.links(3).Tc=coul(3);
110 arm.links(3).Jm=Jm(3);
111 arm.links(3).G=gear(3);
112
113 % link 4
114 arm.links(4).m=0.485;
115 r4=[0.058 -0.018 0.025];
116 G43=(R30'*r4')';
117 T43=arm.links(4).A(qnom(4));
118 G44=inv(T43)*transl(G43);
119 arm.links(4).r=transl(G44)';
120 Ixx=0.001; Iyy=0.002; Izz=0.002;
121 Ixy=0; Ixz=0; Iyz=0;
122 I4=[Ixx Ixy Ixz;
```

```
123      Ixy Iyy Iyz;
124      Ixz Iyz Izz];
125  R40=roty(pi/2)*rotz(pi);
126  arm.links(4).I=R40*I4*R40';
127  arm.links(4).B=fric(4);
128  arm.links(4).Tc=coul(4);
129  arm.links(4).Jm=Jm(4);
130  arm.links(4).G=gear(4);
131
132  % link 5
133  arm.links(5).m=0.19;
134  r5=[0.0532 0.0149 -0.00233];
135  T4=[Lx4 0 0];
136  r5t=r5+T4; % translation
137  G54=(R40'*r5t')';
138  T54=arm.links(5).A(qnom(5));
139  G55=inv(T54)*transl(G54);
140  arm.links(5).r=transl(G55)';
141  Ixx=0.0000952; Iyy=0.000414; Izz=0.00045;
142  Ixy=0.000108; Ixz=-3.08e-5; Iyz=-5.52e-6;
143  I5=[Ixx Ixy Ixz;
144      Ixy Iyy Iyz;
145      Ixz Iyz Izz];
146  R50=roty(pi/2)*rotx(-pi/2);
147  arm.links(5).I=R50*I5*R50';
148  arm.links(5).B=fric(5);
149  arm.links(5).Tc=coul(5);
150  arm.links(5).Jm=Jm(5);
151  arm.links(5).G=gear(5);
152
153  % link 6
154  arm.links(6).m=0.359;
155  r6=[-0.00861 -0.00601 0.0127];
156  G65=(R50'*r6')';
157  T65=arm.links(6).A(qnom(6));
158  G66=inv(T65)*transl(G65);
159  arm.links(6).r=transl(G66)';
160  Ixx=0.000221; Iyy=0.000239; Izz=0.000214;
161  Ixy=-9.73e-6; Ixz=-2.18e-5; Iyz=0.0000222;
162  I6=[Ixx Ixy Ixz;
163      Ixy Iyy Iyz;
164      Ixz Iyz Izz];
165  R60=roty(pi/2)*rotz(pi);
166  arm.links(6).I=R60*I6*R60';
167  arm.links(6).B=fric(6);
168  arm.links(6).Tc=coul(6);
169  arm.links(6).Jm=Jm(6);
170  arm.links(6).G=gear(6);
171
```

```
172 % link 7
173 arm.links(7).m=0.0344;
174 r7=[0.0105 0.000314 -0.000366];
175 T6=[Lx6 0 0];
176 r7t=r7+T6; % translation
177 G76=(R60'*r7t')';
178 T76=arm.links(7).A(qnom(7));
179 G77=inv(T76)*transl(G76);
180 arm.links(7).r=transl(G77)';
181 Ixx=0.000008; Iyy=0.00000642; Izz=6.71e-6;
182 Ixy=-8.56e-8; Ixz=5.86e-8; Iyz=1.22e-7;
183 I7=[Ixx Ixy Ixz;
184     Ixy Iyy Iyz;
185     Ixz Iyz Izz];
186 arm.links(7).I=R60*I7*R60';
187 arm.links(7).B=fric(7);
188 arm.links(7).Tc=coul(7);
189 arm.links(7).Jm=Jm(7);
190 arm.links(7).G=gear(7);
```