Programming of Robotic Manipulation Tasks By Demonstration

Alper Denasi

DCT 2009.034
Abstract

There is a rising trend towards integrating service robots in domestic environments to assist people with their daily tasks. A common feature of those environments lies in the fact that they are dynamic, unstructured and uncertain. Therefore, robust and intelligent programming of service robots becomes a challenging issue. In this master’s thesis, various existing approaches to programming complex robotic manipulation tasks which are related with household applications have been investigated.

The concept of atomic skills which is used to decompose a complete domestic task into smaller subtasks has been introduced. This decomposition has helped to realize that most of these skills require interaction with the environment (i.e. constrained motion tasks) and transitions between free and constrained motions. Therefore, the notion of natural/artificial constraints related to a given task has been introduced for better execution of tasks involving extensive contact with the environment.

An approach to execute constrained motion tasks has been discussed by using natural/artificial constraints. The concept of contact impedance has been introduced to describe the interaction between the manipulator and the environment. It has been used both for identifying the natural constraints of the environment and for tuning of the controller that is used in constrained motion phase.

Another important issue is ability to control the interaction between the robot and the environment. Impedance control has been selected for this purpose, since it can be used both in free and constrained motion phases. A passivity based approach has been used, since it offers some robustness against unmodeled manipulator and actuator dynamics. Two different types of impedance controllers have been implemented both in simulations and experiments. The first one is implemented in joint coordinates, whereas the second one is implemented in Cartesian coordinates.

The contact impedance identification algorithm has been tested both in simulations and experiments. Three different experiments have been conducted on two different objects. The objects used during the experiments are a stiff object, a liftboy, with a flat surface and a soft object, a sponge. All considered identification experiments and simulations contained three consecutive phases: unconstrained, constrained and again unconstrained motion. It has been realized from experiments that the linear environment model used in the identification algorithm is not sufficient to describe the behavior of the soft object and the effect of contact friction.

By using the introduced approach, very simple demonstrated tasks have been replayed, similar to the ones that are used in the identification simulations and experiments.
Acknowledgements

This thesis presents the outcomes of my graduation project which has been performed at the Dynamics and Control Group of the Eindhoven University of Technology. The project has been carried out under the authority of the Mechatronics department of Philips Applied Technologies. I have received a lot of help and support from many people during this project.

First of all, I would like to thank professor H. Nijmeijer for providing the opportunity to work on this project and for all the valuable comments during our monthly discussion meetings. Secondly, I want to thank my coaches Dragan Kostic and Boudewijn Verhaar for all their help, support and critical comments throughout the entire project. Thirdly, I want to thank the people and students at the Mechatronics department of Philips Applied Technologies for many interesting discussions which we had. Finally, I would like to thank my family, especially my parents for their never ending support and confidence in me during my entire study period.
Contents

ABSTRACT ......................................................................................................................................... I

ACKNOWLEDGEMENTS ............................................................................................................... II

1. INTRODUCTION ......................................................................................................................... 1
  1.1. Motivation and Background ................................................................................................. 1
  1.2. Problem Statement ............................................................................................................... 1
  1.3. The Robot Control System ................................................................................................. 2
  1.4. Outline .................................................................................................................................. 3

2. LITERATURE SURVEY .............................................................................................................. 5
  2.1. Atomic Skills ......................................................................................................................... 5
  2.2. Pre-programming of Tasks ................................................................................................. 6
  2.3. Programming by Demonstration (PbD) ............................................................................... 7
    2.3.1. Trajectory level learning ................................................................................................. 7
    2.3.2. Symbolic level learning ................................................................................................. 10
    2.3.3. Environment characterization ......................................................................................... 10
  2.4. Conclusions .......................................................................................................................... 12

3. IMPEDANCE AND CONSTRAINT IDENTIFICATION .............................................................. 15
  3.1. Natural and Artificial Constraints ....................................................................................... 15
  3.2. Contact Impedance ............................................................................................................. 17
  3.3. Identification of the Linear Contact Model ......................................................................... 18
    3.3.1. Weighted Least Squares Estimation ............................................................................. 20
    3.3.2. Uncertainty Estimation ................................................................................................. 20
    3.3.3. Forgetting factor depending on Speed of Movement ................................................... 22
    3.3.4. Discontinuity detection ................................................................................................. 23
    3.3.5. Batch Least Squares ..................................................................................................... 24

4. IMPEDANCE CONTROL .......................................................................................................... 27
  4.1. Free Motion Control ............................................................................................................. 28
    4.1.1. Joint Space Control ....................................................................................................... 28
    4.1.2. Cartesian Space Control ............................................................................................... 31
  4.2. Constrained Motion Control ............................................................................................... 33
  4.3. Contact Transition Control ................................................................................................. 37
  4.4. Summary .............................................................................................................................. 38
1. Introduction

1.1. Motivation and Background

In the coming years, consumer robots will be developed to make lives of humans more enjoyable. These robots will for example entertain humans, or assist them with daily tasks. Such tasks have to be completed successfully in environments which are both unknown a-priori and subject to change, such as in house-holds, offices, outdoor, etc. These tasks should be dealt with robustly and efficiently, bearing the safety of the humans in the mind. Examples of such domestic applications are,

- kitchen tasks (cleaning the worktop, laying the table, picking and placing objects from cupboards, drawers, or cutting vegetables, etc.),
- garment care (sorting, ironing and folding clothes),
- helping handicapped people getting up from their bed, and so on.

Since the domestic environments are unstructured, dynamic and uncertain, completing these tasks successfully in these environments becomes a challenging issue for domestic robots as opposed to industrial robots. The working environments of industrial robots are well defined and the robot interacts with a very limited number of objects. This makes the design and programming of the controllers for industrial robots much easier when compared to domestic robots. Thus industrial robots can obtain high performance in terms of speed, accuracy and reliability. Such a high performance is not necessary for operation of domestic robots; robustness is of the main interest. Pre-programming of tasks is very tedious or impossible, due to all possible exceptions that might occur since every environment is different and non-stationary. Therefore, a different way of ‘programming’ tasks is required in order to cope with this issue.

1.2. Problem Statement

The research problem can be defined as below.

There is an increasing interest in the construction and intelligent programming of robotic arms that can be used for developing and demonstrating a variety of domestic tasks. Typically, these arms are designed to be lightweight and of low stiffness for safe operation in domestic environments. As a consequence, the motions of the various degrees of freedom of the arm will have a strong interaction.

A common approach to realize the complex nature and the variety of domestic tasks is by subdividing them in the smallest required number of atomic skills (also called subtasks). The complexity of the complete domestic task can be reduced in this way. These atomic skills require a robust combination of sensor reading, sensor fusion, state estimation, decision synthesis, trajectory planning, and control. Atomic skills need to be programmed or learned for a successful operation of the robot. Programming may be very extensive and time-consuming; therefore, learning may be an interesting alternative.
Learning of atomic skills by human demonstration is considered in this master thesis. In particular, the following issues are investigated:

1. Domestic tasks are complex and diverse in their nature. They involve the use of complex manipulation skills, such as grasping, pushing, transporting, etc. These skills are more challenging compared to tasks such as setpoint regulation or trajectory tracking. Therefore, knowledge about existing methods and potential approaches for programming these complicated tasks in robotic systems should be developed.
2. Training and/or learning based approaches for programming the aforementioned atomic tasks can in principle greatly reduce the programming effort. This is especially important for domestic end-users. Thus, a literature study of the related approaches should be conducted and the resulting know-how should be presented.
3. A suitable approach to learning an atomic skill should be selected and implemented experimentally on the existing robotic arm set-up, so that the effectiveness of the approach can be tested under real circumstances.

Therefore the main objective of this master thesis assignment is to find a suitable way to ‘program’ the atomic skills of domestic robots efficiently and robustly.

### 1.3. The Robot Control System

The schematic of a typical robot control system is presented below in Figure 1.1. This block diagram can give a good overview of the fundamental components of the robot control system.

![Overall diagram of a robot control system](image)

The task planning algorithm is responsible for making the necessary decisions to solve the task defined by the user (or operator). The objective of the task can be provided to the system by using a suitable user interface. This task specification is then transformed by the planning system into several subtasks (also called atomic tasks or task primitives). The planning system can make use of position (encoder, etc.), force (strain gage, load cell, etc.) and vision type (i.e. camera) sensors. Vision type sensors are not used in this thesis, and focus is on tasks that can be accomplished blindly.

The setpoint generator module calculates the continuous trajectories (positions, velocities, forces, etc.) including the limitations coming from the robot’s kinematics and dynamics, and also constraints from the environment.
The controller module generates the necessary control inputs (also called signals or actions) which are sent to the robot’s actuators to be executed.

1.4. Outline

The report proceeds as follows. The second chapter of this thesis discusses several pre-programming and programming by demonstration methods that are used in the field of artificial intelligence and robotics. The third chapter discusses an approach based on constraint and impedance identification. An impedance control method for manipulators with low joint stiffness (i.e. flexible joint robots) is presented in the fourth chapter. In chapter five, simulation results related to contact impedance identification and impedance control are introduced. The experimental results related to impedance identification and impedance control are given in chapter six. Finally, the conclusions and recommendations are given in the last chapter.
2. Literature Survey

This chapter introduces existing methods for programming and training complex and diverse tasks and atomic skills used in the robotics literature. Robotic tasks and atomic skills can mainly be accomplished in two ways, either by pre-programming or by human demonstrations. Both methods are frequently used to solve complex tasks where the latter concerns home users and domestic environments more. In this chapter, first atomic skills will be introduced together with an example. Then further examples from selected papers will be introduced for both pre-programming and programming by demonstration of robotic tasks and atomic skills.

2.1. Atomic Skills

Atomic skills have a single clear physical objective and cannot be physically further separated into multiple meaningful sub-objectives. A complete domestic task will consist of a sequence of multiple atomic skills; leading to a hierarchical representation of that task. An example of such task decomposition is given for the cutting carrot task as below.

1. Cutting a carrot
   1.1. Preparations
      1.1.1. Place cutting board flat on worktop
      1.1.1.1. Find rough position of cutting board (vision input) and rough orientation (upright or flat)
      1.1.1.2. Move cutting board (depends on determined position of cutting board):
         1.1.1.2.1. Upright:
            1.1.1.2.1.1. Grab both sides of cutting board with both arms simultaneously, with sufficient normal force
            1.1.1.2.1.1.1. Open fingers of both hands
            1.1.1.2.1.1.2. Move hands towards sides of cutting board (with correct orientation)
            1.1.1.2.1.1.3. Move hands sideways towards cutting board until touching
            1.1.1.2.1.1.4. Close fingers
            1.1.1.2.1.1.5. Apply sufficient normal force
         1.1.1.2.2. Flat:
            1.1.1.2.2.1. ..........  
            1.1.1.2.2.1.1. .......... 
            1.1.1.2.2.1.2. .......... 
         1.1.1.2.2.2. .......... 
   1.1.2. Wash carrot
      1.1.2.1. Find position and orientation of carrot (vision input)
      1.1.2.2. Pick up carrot with hand 1 (assumption: carrot is lying flat)
         1.1.2.2.1. Open fingers of hand 1
         1.1.2.2.2. Move to rough position of carrot (fingers pointing down)
         1.1.2.2.3. Move down until fingers touching surface next to carrot
1.1.2.2.4. Slide fingers over surface towards carrot until touching
1.1.2.2.5. Apply normal force to the carrot (sufficient but not too much)
1.1.2.2.6. Lift (movement in vertical direction) (monitor slip to make sure to apply enough normal force for friction)
1.1.2.3. Open tap with hand 2 (while holding carrot with hand 1)
   1.1.2.3.1. Open fingers
   1.1.2.3.2. ............
1.2. Cut carrot
   1.2.1. Pick up knife from cutting block
       1.2.1.1. ............
       1.2.1.2. Hold the handle of the knife
           1.2.1.2.1. Open fingers of hand 1
           1.2.1.2.2. Move to rough position of knife handle
           1.2.1.2.3. Close fingers of hand 1
           1.2.1.2.4. Pull the knife out of the cutting block
       1.2.1.3. ............
   1.2.2. Cut carrot
       1.2.2.1. ............
       1.2.2.2. ............
   1.2.3. ............

Here the underlined entries represent the aforementioned atomic skills and the dots represent the further subdivision of the task. After these atomic skills are taught or preprogrammed, they can be combined to create more complete and also more complex and different tasks. A common feature that is often encountered in this hierarchical representation is that there are many transitions between force and position tasks.

2.2. Pre-programming of Tasks

Robotic tasks can be pre-programmed either by using low-level or high-level methods using robot programming languages [18]. According to Meeussen, low level approaches can be specified in three ways either at the joint level, or at the end-effector level or at the object level (i.e. according to the manipulated object). In the joint level specification, a sequence of positions and velocities of the robot’s joints is recorded with the help of a teaching pendant (or a keyboard) in order to be replayed by the robot during execution. Position and orientation of the robot’s end-effector are recorded by using a teaching pendant (or a keyboard) in the end-effector level specification. The operator specifies spatial relationships between objects, based on a geometric model of the objects in the object level approaches. This information is then used in the programming language to determine the required position and orientation of the manipulated object together with the robot’s end-effector. High level approaches are comprised of commands such as inserting an object into another one, grasping an object, moving it to a specified position, etc. These commands are specified by the operator without explicitly giving any information about how to execute the task, thus leaving it to the task planning algorithm to produce low-level commands for the robot to execute.

A similar way of classification has also been done by Cutkosky et. al. with an additional middle level [8, 9, 10]. This middle level contains phases, transition control and event detection in order to create a bridge between high and low level approaches. Phases control the operation during a task segment (e.g. free motion/constrained motion) in which a
particular set of constraints is active and events (which are discrete entities such as contact, slip detection, etc.) signal the shifts from one phase to the next. Their aim is to create a general purpose programming environment to facilitate programming of complex dexterous manipulation tasks. One of their main contributions is the robust and reliable event detection algorithm that they have developed.

Williamson used Matsuoka oscillators in order to control the joints of a robot arm by exploiting the natural dynamics of the arm and its environment [15]. He demonstrated the approach experimentally on a robotic system containing two 6 d.o.f. arms with series elastic actuators on dynamic tasks such as tuning into the resonant frequencies of the arm itself, juggling, turning cranks, playing with a Slinky toy, sawing wood, throwing balls, hammering nails and drumming. This method is not suitable to perform tasks that require precise manipulation such as sewing, writing or peeling vegetables.

2.3. Programming by Demonstration (PbD)

Robot programming by demonstration (more biologically called imitation learning) can be an efficient and easy method for unskilled users to integrate skills to robots. According to Billard et. al. programming by demonstration methods can generally be classified into two categories, trajectory encoding which is a low-level learning of skills and symbolic encoding, a high-level learning of skills [1]. A similar way of skill learning classification is also performed by Dillmann et al. In addition, they also introduced a classification of how to demonstrate examples to a robotic system being either active or passive or implicit [19]. Tasks are performed directly by the user with the help of data-gloves, cameras, and haptic devices in the case of active examples. Passive examples are introduced by means of master-slave systems, and implicit examples are related to selecting a set of graphical commands to define the learning system’s goals.

There are many methods for teaching low-level skills which are also called as elementary (or atomic) skills such as machine learning methods, or using dynamical systems. Among the machine learning tools, there are artificial neural networks (ANN), radial basis function networks (RBF), hidden markov models (HMM) and Gaussian mixture models (GMM). Learning skills by using dynamical systems can be done by using nonlinear attractors (or oscillators) or by using recurrent neural networks (RNN). High-level learning of skills from human demonstrations is often comprised of learning basic action (or skill) sequences in a hierarchical manner [1]. Hidden markov models are one way of encoding and regenerating these sequences [1].

Other than these two approaches, there are also methods to characterize or identify the properties of the environment that the robot interacts with such as stiffness, viscosity and inertia. These properties can then be used in stiffness or impedance based controller schemes that are used to control the robots.

2.3.1. Trajectory level learning

Trajectories obtained from human demonstrations are saved in terms of either joint variables or task variables [1]. Research on imitation learning specified three types of motions, either
repetitive (i.e. circular movements as in table wiping) or discrete motions (i.e. point to point movements) or a combination of both.

Calinon et. al. used GMM in order to encode a set of trajectories containing either joint angles, or hand paths, or hands-object distance vectors [2]. Principal component analysis (PCA), a dimensionality reduction technique, has been used firstly in order to represent the data obtained from sensors on an optimal latent space. An example of an optimal latent space is, for a writing task, the projection of the 3 dimensional Cartesian position of the hand on the 2 dimensional writing surface. Then they used Gaussian mixture regression to obtain the generalized version of the processed trajectories. They demonstrated the validity of their approach by three different experiments on a humanoid robot. Those experiments are comprised of grabbing and moving a chess piece on a chess board, grabbing a bucket with two hands and moving it to a specific location, and grabbing a piece of sugar and bringing it to its mouth with either left or right arm. The advantages of the approach are,

• Important qualitative features of each task, being high-level goals, are not explicitly represented in the robot’s control system.
• Generalization for different initial object positions is possible on the task’s latent space.

The disadvantage of this method can be given as,

• Kinematics information has been implicitly assumed to be sufficient to describe a task and dynamics information (such as forces applied on an object) has been neglected.

This approach has further been developed by using an incremental learning approach in which the demonstrator observed the outcomes of the robot’s execution and then prepares the next examples according to the robot’s success [7]. This resulted in learning the task cooperatively by putting the teacher in the robot’s learning loop. They experimented on two cases, where the first task is to grasp and move a large foam die bimanually, and the second is to move chess pieces on a chess board.

Nechyba et. al. used feedforward neural networks with the cascade two learning architecture in order to learn the human control strategies [4]. They experimented on balancing an inverted pendulum-cart system on a computer screen where a human subject can control the horizontal forces applied to the cart via the horizontal mouse position. The advantages of the approach are,

• The architecture of network is adjusted automatically, prior guessing is not necessary.
• It can model higher degrees of nonlinearity with fewer hidden units than might be required.
• Incremental addition of hidden units allows for new hidden units to have variable activation functions.

The disadvantage of the method is,

• The function approximator can trade-off some generalizability in favor of better approximation.

Kaiser et. al. used radial basis function networks in order to acquire elementary skills from demonstrations [5]. They define a skill as a control function,
\[ C_x : \mathbf{u}(t) = C_x(\mathbf{x}(t)) \]  

where \( \mathbf{x}(t) \) is the state of the system (robot) and \( \mathbf{u}(t) \) is the action that the robot should apply to fulfill a goal. The state is defined as a sequence of sensorial inputs \( \mathbf{y}(t) \), 

\[
\mathbf{x}(t) = (\mathbf{y}(t-d), ..., \mathbf{y}(t-d-p)), \quad \text{where} \quad d, p \geq 0.
\]

They use measured forces/torques as sensorial inputs and translational/rotational velocities as actions. The goal is represented by a reward function \( r(\mathbf{x}(t)) \) according to the distance of the current state to the desired state. Then, they preprocess the data in order to determine relevant action components, and relevant perceptions and remove incorrect actions that do not contribute anything to solving the task, by applying thresholding to the action values. Two different manipulation skills have been experimented with their approach, first inserting a peg into a hole, second opening a door. The advantages of the method are,

- The method is generic since it is a function approximation problem.

The disadvantage of the approach are,

- A reward function should be defined which can be difficult for different kinds of tasks.

Learning trajectories by using dynamical systems is a way to exploit the dynamics of the movements are used in [11, 12, 13]. According to Billard et. al. learning trajectories by dynamical systems has the advantages given below,

- The learning algorithm used in this approach does one-shot learning, and avoids slow convergence of many neural network algorithms.
- They are reusable since they represent the learned trajectories with a few online modifiable parameters, which is very important for generalization of movements.
- Their design gives them intrinsic robustness against perturbations.
- They can be used for movement classification, thus providing a tool for measuring similarities and dissimilarities between trajectories.

Urbanek et. al. used this approach for solving table wiping problem [14]. The basic circular movement (i.e. movement primitive) is taught by using demonstrations obtained through guiding the robot arm by using its end-effector. The complete movement plan is obtained by using simulated annealing with the A* global search algorithm. The advantage of this approach is,

- A way to combine the movement primitives, (i.e. the complete solution of the task) is introduced.

The drawbacks of the approach are,

- It is limited to tasks containing repetitive movements.
- It cannot distinguish in which directions force control is more important than position control and vice versa.
2.3.2. Symbolic level learning

Robotic tasks can also be learned by segmentation according to predefined actions. Hovland et. al. used a hidden markov model to model an assembly skill by observing human demonstrations [17]. Then they used a discrete event controller to execute the identified transition probabilities from HMM. They verified their approach experimentally on a peg insertion task. The advantage of their approach is,

- Unlike methods like neural networks, the acquired skill can be described physically meaningfully based on HMM parameters.

The drawback of the method is,

- It may be possible that the system may not handle situations that are not present in the training set.

2.3.3. Environment characterization

The tasks that the robots will perform in domestic environments will have a lot of interactions with such environments. Acquiring properties of the environments that could also be considered as haptic properties can be beneficial in executing demonstrated tasks. Karon E. MacLean created an automated haptic characterization technique, where a tight position controlled haptic display probed a real environment with a nonlinear, compliance-dominated impedance, while measuring the interaction forces [6]. It is also possible to identify other impedance properties during contact with the environment such as mass \( M(x) \) and damping \( B(x) \) using this method. The experimental setup that has been used in this approach is depicted as in Figure 2.1.

![Figure 2.1 Schematic of experimental setup used in [6].](image)

The force probe moves from left to right and pushes the toggle switch till its maximum deflection occurs during experiment phase. The environment impedance used in the experiments is assumed to be piecewise linear as a function of displacement \( K(x) \). The advantages of this approach are,

- The identification procedure of the model parameters, including the data collection and analysis, is rapid.
- The algorithm can also be used to identify nonlinear characteristics such as stiction, etc.

The drawback of this approach can be given as,
• The method identifies the parameters of an assumed model structure (i.e. piecewise linearity assumption), it does not create a model structure by itself. This means that if the assumption is not representative of the reality well, the algorithm can misfit the impedance components.

In Sikka et. al., contact tasks are modeled according to the end-point stiffness calculated from human demonstration data (i.e. recorded position and force signals) [16]. They use a generalized spring to model the relation between the object being manipulated and the environment. Their assumption is that the relation between the object manipulated and the environment can be given by

\[ F(t) = K(t) \left[ X_d(t) - X(t) \right] \]  

(2.2)

where \( F(t) \) denotes force vector acting on the object, \( X_d(t) \), the desired position of the object, \( X(t) \) referring to the actual position of the object and \( K(t) \) is the stiffness matrix. These notations are introduced in the Figure 2.2.

The data obtained from demonstrations is characterized according to 4 situations; the object is stationary in free space, or it is in contact with the environment, or it is moving in free space, or it is moving in contact with the environment (i.e. constrained motion).

The advantage of this method can be given as below,

• The entire demonstration does not need to be segmented into distinct subtasks and to be executed by switching control laws afterwards.

The disadvantage of this method is,

• The demonstration data should be smoothed before the desired trajectory and stiffness matrix obtained with this method can be executed.

Kikuuwe developed an identification technique of constraint condition based on position and force sensing during arbitrary manipulation [3]. The impedance parameters are estimated online together with detection of discontinuous changes in the estimated impedance. It is assumed that the force-motion relation can be locally approximated by a linear dynamic equation in a local region as below,

\[ f(t) = c + K p(t) + B \ddot{p}(t) + M \dot{p}(t) \]  

(2.3)
where \( p(t) \) is position of robot’s end-effector and \( f(t) \) is force applied from the end-effector to the environment. \( K \in \mathbb{R}^{3 \times 3} \), \( B \in \mathbb{R}^{3 \times 3} \), and \( M \in \mathbb{R}^{3 \times 3} \) are stiffness, viscosity and inertia matrices respectively, determined by dynamic properties of the end-effector and the environment, while \( c \) is a constant vector corresponding to equilibrium position of stiffness, bias of force resulting e.g. from gravitation. Two experiments have been performed, where in the first experiment the ability to detect changes in stiffness has been tested, whereas in the second one the ability to distinct constraint directions has been investigated.

The stiffness matrix obtained from this impedance perception method, has been utilized further for extracting information about the properties of flat and cylindrical surfaces [20, 21]. The experiments on the flat surface are performed for two different cases; horizontal and inclined postures. The normal direction, the stiffness along the normal direction and the friction coefficient of the surface is obtained through these experiments. The primary directions, the normal direction, curvature, stiffness and friction coefficients of the cylindrical surface are obtained by conducting three different surface following experiments.

### 2.4. Conclusions

Robot programming methods include designing both low-level and high-level layers of the task that will be solved. The task planner introduced in chapter 1.3, will be responsible for executing this hierarchy. Based on this hierarchical separation, the first important aspect of a robot program is to design the lower level. This task decomposition can either be performed manually for each domestic task, or can be learned in case of programming by demonstration. According to the atomic skill (subtask or elementary action) breakdown introduced in chapter 2.1, manipulation tasks can be separated into unconstrained motion and constrained motion segments. In a complete robotic task, there will be many different objects with different impedance (e.g. soft, rigid) properties and various subtasks that require interaction with these objects in different contact conditions during constrained motion phases. Therefore, the constrained motion phase should further be separated into more meaningful segments, taking this information into account. Environment characterization techniques introduced in chapter 2.3, can be one way to achieve this purpose. The estimator which will be used in the environment characterization (also called environment property estimation) process should be able to work both offline for subtasking purposes, and also online in order to detect the variations in a dynamic environment. An important feature of the environment characterization techniques is that, they can also facilitate the design of the controllers (force control, impedance control, etc.) in the constrained motion case. A limitation of these methods is that they make an assumption about the model of the environment, which can lead to the misinterpretation of the skill, if the model deviates a lot from the reality. Secondly, for the robot control system to work properly, the trajectories in each subtask should be designed in the case of manual programming, or should be generated automatically with path planning algorithms (for autonomous execution) or should be learned by the trajectory learning methods introduced in chapter 2.3.1 for programming by demonstration purposes. Trajectory learning methods can be beneficial for robots working in domestic environments, since there will be many different trajectories in a service scenario such as the one introduced in chapter 2.1 and it will be quite cumbersome to design each trajectory manually. Such a trajectory learning method should be able to parameterize the demonstrated trajectories with as low number of parameters as possible. Furthermore, it should be able to cope with sensor noise and human motion inconsistencies (also called demonstration noise) and preferably it should
be able to learn from a single demonstration. Among the trajectory learning methods, using dynamical movement primitives can be an appealing approach, since the learning is performed by using only a single demonstration, and reproduction is done by changing only very few parameters. A limitation of this method can be however, for different type of trajectories different kind of primitive (rhythmic, discrete) might be necessary. So they should be used together with a movement classification algorithm. GMM can be another way to teach trajectories to the robot from human demonstrations. The method can deal with human noise; however it is unclear whether it can learn a trajectory from a single demonstration.
3. Impedance and Constraint Identification

Robotic tasks that contain interaction with the environment, such as cleaning a table, inserting a peg into a hole, or opening a door, involve extensive contact. These tasks are called constrained motion (or compliant motion) tasks in general and can be accomplished by a proper specification of compliant elementary actions. In this chapter, first the concept of physical constraints of the environment is introduced, then contact impedance is explained and finally an impedance identification method is presented. Before going into further detail about these, a proper definition of constrained motion is introduced.

**Definition 3.1**

1. If \( \{e_1, \ldots, e_6\} \) is a basis for the vector space \( \mathcal{M} \), and \( \{f_1, \ldots, f_6\} \) is a basis for \( \mathcal{F} \), we say that these basis vectors are **reciprocal** provided

\[
e_i^T f_j = \begin{cases} 0, & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \tag{3.1}
\]

2. An instantaneous rigid body linear and angular motion (also called twist,
\( \xi = \begin{bmatrix} \mathbf{v}^T \\ \omega^T \end{bmatrix} \in \mathcal{M} \)) which is executed by a rigid manipulated object is **reciprocal** to the ideal (i.e. excluding friction) reaction forces and moments (also called wrench,
\( F = \begin{bmatrix} f^T \\ n^T \end{bmatrix} \in \mathcal{F} \)) that the contact condition can generate [22, 37]:

\[
\xi^T F = v^T f + \omega^T n = 0 \tag{3.2}
\]

### 3.1. Natural and Artificial Constraints

In this section the concept of constraints are introduced together with a few representative examples by using the approach called compliance (also called constraint or task) frame modeling. Furthermore an example task that cannot be modeled by using this concept is also introduced. The task frame method facilitates the description of the interaction task by allowing the user to define six programmable directions for position/force control, related to the contact situation. The natural constraints in either position or force are related to the task geometry, whereas the desired position or force setpoints provided by the user to accomplish the task are called artificial constraints. Bruyninckx et. al. in [22] introduce three requirements for the applicability of the approach, together with three relaxing options for the practical use of the method. These are only going to be mentioned very briefly here. The requirements include, how to model the constraint frame in a geometrically compatible way, how to select the relevant control actions (position/force setpoints, controller gains, etc.) based on this model and finally how to adjust these according to continuously changing contact conditions during motion. The options can be summarized as: 1) the constraint frame should be placed optimally to provide maximal decoupling for position/force controlled directions, 2) if the controller is robust enough to handle task execution errors, then a simpler task frame can be selected in lieu of geometric compatibility, and 3) the control actions (force/position control) can be selected according to the expected contact condition instead of the current one when
specific controllers (e.g. damping control) are selected. The first example task is related to inserting a round peg into a round hole with a desired velocity \(v_{\text{des}}\). The compliance frame for this task is depicted in Figure 3.1.

![Figure 3.1 Inserting a peg into a hole [22]](image)

It is assumed that both the walls of the hole and the peg are perfectly rigid, and the contact is frictionless. Based on this assumption, it can be observed that the peg cannot move in X and Y directions and also cannot rotate around X and Y directions. The peg’s diameter is not the same as for the diameter of hole, there is sufficient clearance (tolerance) such that the peg can enter inside the hole. It can only move in Z and rotate around Z directions. On the other hand it can be observed that the peg cannot exert force or apply moment in Z direction. These constitute the natural constraints related to this task which can also be viewed in Figure 3.1. The remaining values, being the artificial constraints, can be arbitrarily assigned for the control system to execute the task. The second example is concerned with opening a door by grasping it from its handle which often occurs in domestic task scenarios. The corresponding task frame for this task is presented in Figure 3.2 as below.

![Figure 3.2 Opening a door [22]](image)

The kinematics of this task restricts the motion of the system to a circular trajectory around
the frictionless door hinge. The constraint frame is attached to the door hinge whose Z axis coincides with hinge’s axis of rotation. This means that it is not possible to apply moments around the Z axis since the hinge is assumed to be frictionless. According to this constraint frame, the natural and artificial constraints related to this task can be viewed in Figure 3.2.

The last example is related to a task that contains multiple contact points, with which the considered approach cannot cope. This task is concerned with following a trajectory along the seam while maintaining contact with both planes at both contact points. By examining this task it can be observed that a single constraint frame is not sufficient to model this task (especially if the plates connected by the seam are not perpendicular). A few similar examples pointing out this incapability of the compliance frame method are also indicated in [23].

Further information about how to model physical constraints of the environment for different kind of tasks can be found in [22], [23].

3.2. Contact Impedance

The reciprocity definition introduced at the beginning of this chapter may not suit very well to some real situations because of contact deformation and friction. The notion of mechanical impedance can be useful to represent the dynamic relation between force and motion in order to describe the interaction between the robot and the environment. There are two types of contact impedance (viscoelastic) models in the literature, which are used in order to describe the contact dynamics: the linear Kelvin-Voigt and the nonlinear Hunt Crossley model. The linear contact model was selected and implemented in this work because of its simplicity. The latter is also more suitable for describing the interaction behavior of soft materials [35]. The Kelvin-Voigt environment model is represented for a single degree of freedom system as follows,

$$f(t) = \begin{cases} K x(t) + B \dot{x}(t), & x(t) \geq 0 \\ 0, & x(t) < 0 \end{cases}$$  (3.3)

where $x(t)$ represents the penetration within the environment, $K$ and $B$ represent the stiffness and the damping of the environment respectively. The Hunt-Crossley model is represented for a single degree of freedom as follows,
\[ f(t) = \begin{cases} K x^o(t) + B x^o(t) \dot{x}(t), & x(t) \geq 0 \\ 0, & x(t) < 0 \end{cases} \quad (3.4) \]

where \( x(t) \) represents the penetration inside the environment, \( K \) and \( B \) represent the stiffness and the damping of the environment respectively, and \( n \) is a constant parameter, typically around 1, which depends on the geometry and material of the contacting objects [35]. The nonlinearity of this model arises from the fact that the spring restoring force \( K x^o(t) \) is nonlinear when \( n \neq 0 \) and the damping force \( B x^o(t) \dot{x}(t) \) is also position dependent.

### 3.3. Identification of the Linear Contact Model

During both the simulations and experiments presented in this thesis, one of the environment characterization methods, proposed by [3], is used. The details of the algorithm are presented in this section. The impedance perception algorithm assumes that the dynamic characteristics of the environment can be approximated locally (i.e. as a single point contact) by a linear dynamic equation. According to the local linearity assumption, the relation between end-effector forces can be given as

\[ f(t) = c + K p(t) + B \dot{p}(t) + M \ddot{p}(t) \quad (3.5) \]

where \( p(t) \in \mathbb{R}^3 \) is the position of robot’s end-effector and \( f(t) \in \mathbb{R}^3 \) is the force applied from the end-effector to the environment. \( K \in \mathbb{R}^{3\times3} \), \( B \in \mathbb{R}^{3\times3} \), and \( M \in \mathbb{R}^{3\times3} \) are stiffness, viscosity and inertia matrices respectively, determined by the dynamic properties of the end-effector and the environment, while \( c \in \mathbb{R}^3 \) is a constant vector corresponding to equilibrium position of stiffness, bias of force resulting e.g. from gravity. However the model given by (3.5) may not represent the contact between the rigid (or nearly rigid) end-effector and rigid (or nearly rigid) environment very well. This is because the force varies in rigid contact case, but the position variation cannot be measured accurately by the typical low-resolution encoders used in the industry. Therefore, it was supposed that the end-effector is virtually visco-elastic, and that is why its virtual position is obtained by adding a simulated displacement to the real position. The force measurements \( f(t) \in \mathbb{R}^3 \) are recorded by using a force sensor. Then, the displacement of the virtual soft effector, \( \Delta p_v(t) \in \mathbb{R}^3 \), is obtained by simulation by solving the following dynamic equation

\[ f(t) = K_v \Delta p_v(t) + B_v \Delta \dot{p}_v(t) \quad (3.6) \]

where \( K_v \) and \( B_v \) are design parameters that represent the stiffness and viscosity coefficients of the virtual soft effector. The position of the virtual soft effector (V.S.F.) is determined as follows,

\[ p(t) = p_v(t) + \Delta p_v(t) \quad (3.7) \]
where \( p_r(t) \in \mathbb{R}^3 \) represents the real end-effector position calculated by using forward kinematics with the measured encoder readings. The end-effector position can also be measured directly by using vision (camera) systems. However in that case, the selected vision system should be very accurate and a specific end-effector for the vision system might be necessary to use. Furthermore, a very accurate image processing software will also be necessary with such a sensor system that will increase computational costs. This would in return increase the costs a lot for a domestic robot. The position \( p(t) \) used in (3.5) is replaced by (3.7) such that the aforementioned limitations in the (nearly) rigid contact case can be overcome. Taking the Laplace transform of both sides of (3.5) gives,

\[
L[f(t) - c] = (K + B s + M s^2)L[p(t)]
\]

(3.8)

where \( L[\cdot] \) denotes the Laplace transformation operator. This continuous-time equation was approximated in discrete-time by using the bilinear transformation (Tustin method),

\[
s := \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}
\]

where \( z \) is the Z-transform operator and \( T \) is the sampling time, as below,

\[
Z[f(t) - c] = \left( K + B \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} + M \left( 2 \frac{1 - z^{-1}}{T} + \frac{1}{1 + z^{-1}} \right)^2 \right) Z[p(t)]
\]

(3.9)

\[
Z[f(t) - c] = \left( \frac{L_1 + L_2 z^{-1} + L_3 z^{-2}}{(1 + z^{-1})^2} \right) Z[p(t)]
\]

(3.10)

\[
\begin{bmatrix} L_1 & L_2 & L_3 \end{bmatrix}^T = (T_1 \otimes I_3)[K \ B \ M]^T
\]

(3.11)

\[
T_1 = \begin{bmatrix} 1 & 2/T & 4/T^2 \\ 2 & 0 & -8/T^2 \\ 1 & -2/T & 4/T^2 \end{bmatrix}
\]

(3.12)

where \( Z[\cdot] \) denotes the Z-transform and \( T_1 \otimes I_3 \) denotes the Kronecker product of \( T_1 \) and \( I_3 \).

After (3.10) is rearranged, the discrete-time approximation of (3.5) in input-output form can be given as follows.

\[
\phi_k = \Theta^T \psi_k
\]

(3.13)

\[
\phi_k = f_k + 2f_{k-1} + f_{k-2} \in \mathbb{R}^3
\]

(3.14)

\[
\psi_k = \begin{bmatrix} p_k^T & p_{k-1}^T & p_{k-2}^T \end{bmatrix}^T \in \mathbb{R}^{10}
\]

(3.15)

where \( \psi_k, \phi_k \) and \( \Theta \) represent the regression vector, the model output and the parameter matrix respectively. In this derivation the variables with the subscript \( k \) denote their value at time instant \( kT \). The relationship between the real parameters (stiffness, damping, etc.) and the parameter matrix \( \Theta \) can be given as below:

\[
\Theta = T \Theta_i \in \mathbb{R}^{10 \times 3}
\]

(3.16)
3.3.1. Weighted Least Squares Estimation

Least squares regression (or estimation, fitting) is a common method to identify the parameters of linearly parameterized input output models such as (3.13). A cost function (or performance index) based on the square of the fitting errors can be defined in order to solve this relationship for its parameters. In an extended version of this method the cost function can be weighted so that different parts of the cost can be given different importance.

According to this, a weighted sum-of-products matrix of residual errors (i.e. cost function) at time instant $kT$, resulting from the model (3.13), can be introduced as follows.

$$J_k(\Theta) = \sum_{i=0}^{k} w_{k,i} (\phi_i - \Theta^T \psi_i)(\phi_i - \Theta^T \psi_i)^T$$

$$= \sum_{i=0}^{k} w_{k,i} \phi_i \phi_i^T - \left( \sum_{i=0}^{k} w_{k,i} \phi_i \psi_i^T \right) \Theta - \Theta^T \left( \sum_{i=0}^{k} w_{k,i} \psi_i \phi_i^T \right) + \Theta^T \left( \sum_{i=0}^{k} w_{k,i} \psi_i \psi_i^T \right) \Theta$$

$$= F_k - Q_k \Theta - \Theta^T Q_k + \Theta^T R_k \Theta$$

(3.21)

where $\{w_{k,i}\}_{0\leq i \leq k}$ denotes the weighting sequence at time instant $kT$, and $i_0$ is the time instant at which the calculation starts. The minimum of the residual errors matrix can be calculated by taking the partial derivative of the residual errors w.r.t. the parameter matrix,

$$\frac{\partial J_k(\Theta)}{\partial \Theta} = 2R_k \Theta - 2Q_k = 0$$

$$\hat{\Theta}_k = R_k^{-1} Q_k$$

(3.22)

(3.23)

which represents the estimate of $\Theta$ at time instant $kT$, provided that $R_k^{-1}$ exists. When the definition of $R_k$ is checked from (3.20), it can be observed that it is related to the regressor matrix $\psi_i \psi_i^T$. Thus theinvertibility of $R_k$ can be guaranteed if the input signals are persistently exciting. When $R_k^{-1}$ exists, (3.21) can be rearranged as follows.

$$J_k(\Theta) = \left( \Theta - \hat{\Theta}_k \right)^T R_k \left( \Theta - \hat{\Theta}_k \right) + S_k$$

$$S_k = F_k - Q_k^T R_k^{-1} Q_k$$

(3.24)

(3.25)

3.3.2. Uncertainty Estimation

An important part of the identification procedure is also how to provide persistently exciting signals such that the parameters of the model are estimated correctly without losing stability of the estimates. Persistently exciting signals can be provided by proper reference signal design in case if the robot is computer controlled. However, if the robot is guided manually by
a person performing a manipulation task, this might not be guaranteed. Nevertheless, the accuracy of the estimated parameters can be evaluated by calculating their uncertainties.

Property 3.1 The vectorization operator denoted by \( \text{vec}[X] \), stacks the columns of its argument matrix \( X \in R^{m \times n} \) as, \( \text{vec}[X] = [x_1^T \cdots x_n^T] \in R^{mn} \) where \( x_i \in R^m \) represents \( i^{th} \) column of \( X \).

Property 3.2 The Kronecker product \( \otimes \) and the vectorization operator have the following mathematical properties.

1. \( (A \otimes B)(C \otimes D) = AC \otimes BD \)
2. \( \text{tr}(A^T B) = \text{vec}(A)^T \text{vec}(B) \)
3. \( \text{vec}(ABC) = (C^T \otimes A)\text{vec}(B) \)
4. \( \text{vec}(AB) = (I \otimes A)\text{vec}(B) = (B^T \otimes I)\text{vec}(A) \)

In order to calculate these uncertainties, a difference measure, \( \Delta_k (\Theta) \), between the estimated parameter matrix \( \hat{\Theta}_k \) and the real one \( \Theta \) can be defined as follows.

\[
\Delta_k (\Theta) = \text{tr}\left[ S_k^{-1/2} (J_k (\Theta) - S_k ) S_k^{-1/2} \right] \tag{3.26}
\]

By using the redefinition given by (3.25) and the properties 3.1 and 3.2, (3.26) can be rearranged as below.

\[
\Delta_k (\Theta) = \text{tr}\left[ S_k^{-1/2} (\Theta - \hat{\Theta}_k)^T R_k (\Theta - \hat{\Theta}_k) S_k^{-1/2} \right]
= \text{tr}\left[ ((\Theta - \hat{\Theta}_k) (S_k^{-1/2})^T )^T (R_k (\Theta - \hat{\Theta}_k) S_k^{-1/2}) \right]
= \text{vec}\left[ (\Theta - \hat{\Theta}_k) (S_k^{-1/2})^T \right]^T \text{vec}\left[ R_k (\Theta - \hat{\Theta}_k) (S_k^{-1/2}) \right]
= \left[ (S_k^{-1/2} \otimes I) \text{vec}(\Theta - \hat{\Theta}_k) \right]^T \left[ (S_k^{-1/2} \otimes R_k) \text{vec}(\Theta - \hat{\Theta}_k) \right]
= \text{vec}\left( \Theta - \hat{\Theta}_k \right)^T \left[ (S_k^{-1/2} \otimes I) \right]^T \left[ (S_k^{-1/2} \otimes R_k) \right] \text{vec}(\Theta - \hat{\Theta}_k)
= \text{vec}\left( \Theta - \hat{\Theta}_k \right)^T \left[ \tilde{\Pi}^{-1} \right] \text{vec}(\Theta - \hat{\Theta}_k) \tag{3.27}
\]

With a design parameter \( A (>0) \), the set of \( \Theta \) which satisfies \( \Delta_k (\Theta) \leq A \) can be given by a hyper-ellipsoid as follows:

\[
\text{vec}(\Theta - \hat{\Theta}_k)^T \tilde{\Pi}^{-1} \text{vec}(\Theta - \hat{\Theta}_k) \leq A \tag{3.28}
\]
By using (3.16), and the properties 3.1 and 3.2, (3.27) can be rearranged as below.

\[
D_k(\Theta) = \text{vec} \left[ T \left( \Theta_I - \hat{\Theta}_{I,k} \right)^T \right] \left[ (S_k^{-1})^T \otimes R_k \right] \text{vec} \left[ T \left( \Theta_I - \hat{\Theta}_{I,k} \right) \right] \\
= \left[ \left( I \otimes T \right) \text{vec} \left( \Theta_I - \hat{\Theta}_{I,k} \right) \right]^T \left[ (S_k^{-1})^T \otimes R_k \right] \left[ \left( I \otimes T \right) \text{vec} \left( \Theta_I - \hat{\Theta}_{I,k} \right) \right] \\
= \text{vec} \left( \Theta_I - \hat{\Theta}_{I,k} \right)^T \left( I \otimes T^T \right) \left[ (S_k^{-1})^T \otimes R_k \right] \left( I \otimes T \right) \text{vec} \left( \Theta_I - \hat{\Theta}_{I,k} \right) \\
= \text{vec} \left( \Theta_I - \hat{\Theta}_{I,k} \right)^T \left( I \otimes T^T \right) \left[ (S_k^{-1})^T \otimes R_k \right] \left( I \otimes T \right) \text{vec} \left( \Theta_I - \hat{\Theta}_{I,k} \right) \\
= \text{vec} \left( \Theta_I - \hat{\Theta}_{I,k} \right)^T \left( I \otimes T^T \right) \left[ (S_k^{-1})^T \otimes R_k \right] \left( I \otimes T \right) \text{vec} \left( \Theta_I - \hat{\Theta}_{I,k} \right) \\
\sum_{i=1}^{I} \Pi_{i,k}^{-1} \\
\text{where } \Pi_{i,k}^{-1} \text{ represent the uncertainty of the true parameters (stiffness, damping, inertia). The uncertainties of each element of } \hat{\Theta}_{I,k} \text{ are given by the } i^{th} \text{ element of } \sqrt{\text{diag} \left( A \cdot \Pi_{i,k}^{-1} \right)}.
\]

3.3.3. Forgetting factor depending on Speed of Movement

The coefficient matrices introduced by (3.5) can gradually vary according to the position, due to the non-linear features of real environments. Therefore, the weighting sequence \( \{w_{k,i}\}_{i=0}^{k} \) has to be selected so that old data are not taken into account. This means that the recursive least squares method is used with some sort of a data forgetting factor. According to Kikuuwe et. al. [3], selecting a constant forgetting speed might result in a narrow data distribution in space domain due to low-speed motion, which can be avoided by using a forgetting factor that changes with the speed of the movement given as below.

\[
w_{k,i} = \begin{cases} 
1 - r_k, & \text{if } i = k \\
r_k w_{k-1,i}, & \text{if } i < k 
\end{cases} \\
r_k = 2^{-\Delta u_k} \\
\text{where } \Delta u_k > 0 \text{ can be referred to as distance between instants } kT \text{ and } (k-1)T \text{ evaluated on a scale of forgetting. When the rule given by (3.30) – (3.31) is used, } W_k = \sum_{j=0}^{k} w_{k,j} \text{ increases monotonically and converges to 1 as } k \to \infty. \Delta u_k \text{ is defined as follows so that low-speed motion makes slow forgetting}
\]

\[
\Delta u_k = \min \left( \frac{T}{T_H}, \frac{\|p_k - p_{k-1}\|}{X_H} \right) \\
\text{where } T_H \text{ and } X_H \text{ represent design parameters, which can be referred to as half-lives in time and space domains respectively. The physical meanings of (3.30), (3.31) and (3.32) are that for low speed motion } \|\dot{p}_k\| = \frac{\|p_k - p_{k-1}\|}{T} \leq \frac{X_H}{T_H} \text{, the minimum becomes } \|\dot{p}_k - \dot{p}_{k-1}\| \text{ and results}
\]
in slow forgetting and a small weight on the current observed values, and for high speed motion \( \| \dot{p}_k \| = \frac{\| p_k - p_{k-1} \|}{T} > \frac{X_H}{T_H} \), the minimum becomes \( \frac{T}{T_H} \) and forgetting occurs at a constant rate \( \frac{T}{T_H} \). When the weights defined by (3.30) are used for the matrices \( R_k \), \( Q_k \), and \( F_k \), defined in (3.20), they can be recursively calculated as below.

\[
R_k = r_k R_{k-1} + (1-r_k) \Psi_k \Psi_k^T \quad (3.33)
\]

\[
Q_k = r_k Q_{k-1} + (1-r_k) \Psi_k \Phi_k^T \quad (3.34)
\]

\[
F_k = r_k F_{k-1} + (1-r_k) \Phi_k \Phi_k^T \quad (3.35)
\]

### 3.3.4. Discontinuity detection

A typical manipulation task will often contain a lot of transitions between contact and non-contact states, or even between different constraint directions while in contact. These transitions will cause discontinuities in the measured impedance of the environment. The control policy in each of these segments will therefore be different. Furthermore the detection of these transitions can help the task planner mentioned in chapter 1 to execute the sequence of subtasks in a more organized fashion. Two different methods are used in this work in order to detect these discontinuities, where the first one is thresholding based on the force sensor signals, and the second one is based on the abrupt changes that occur in the prediction error when there is a transition. The first one is a simple approach that can distinguish between contact and non-contact phases, by checking whether the norm of the force signals exceed a threshold value. The second way to detect discontinuous changes of the constraint condition is by checking whether the current observed model data match a past estimated data. In order to determine an old estimate, at every time instant \( k \), a referential past instant \( \kappa \) can be determined by executing the following summation loop,

\[
U = \sum_{i=\kappa+1}^{k} \Delta u_i \quad (3.36)
\]

where \( U \) is a design parameter (i.e. a threshold) indicating when to stop this loop. At each time instant of the estimation, the summation loop (3.36) is started from the present time instant \( k \), down to the past time instant until the threshold, \( U \) is satisfied approximately. The estimate of the model output, \( \Phi_k \), based on the old model at time instant \( \kappa \), is defined as

\[
\hat{\Phi}_{4\kappa} = \hat{\Theta}_\kappa \Psi_k. \quad \text{From (3.13), the difference between the current observed and the past estimated vectors, based on properties 3.1 and 3.2, can be given as}
\]

\[
\tilde{\Phi}_{4\kappa} = \Phi_k - \hat{\Phi}_{4\kappa} = (\Theta - \hat{\Theta}_\kappa)^T \Psi_k
\]

\[
\tilde{\Phi}_{4\kappa} = (I_3 \otimes \Psi_k^T) \text{vec}(\Theta - \hat{\Theta}_\kappa) \quad (3.37)
\]

where \( \hat{\Theta}_\kappa \), represents the estimate of the parameter matrix at past instant \( \kappa \). Since the uncertainty of this past estimate also satisfies the condition (3.28), it can be rearranged as,
\[
D_{\dot{\kappa}k}(\Theta) \leq A \
\text{vec}(\Theta - \hat{\Theta}_\kappa) \text{vec}(\Theta - \hat{\Theta}_\kappa)^T \leq A\hat{\Pi}_\kappa
\]  

(3.38)

can be stated for the old estimates. By using (3.37), (3.38) can be rewritten as,

\[
\tilde{\phi}_{\dot{\kappa}k} \tilde{\phi}_{\dot{\kappa}k}^T \leq A \left( I \otimes \psi_k^T \right) \Pi_k \left( I \otimes \psi_k \right)
\]

\[
= A \left( \psi_k^T R_k^{-1} \psi_k \right) S_k
\]

(3.39)

(3.40)

After rearranging the terms of this inequality, it can be represented as

\[
e_{\dot{\kappa}k} = 1 - \frac{\tilde{\phi}_{\dot{\kappa}k}^T S_k^{-1} \tilde{\phi}_{\dot{\kappa}k}}{A \psi_k^T R_k^{-1} \psi_k} \leq 1
\]  

(3.41)

It is assumed that when \( |e_{\dot{\kappa}k}| > 1 \) holds at all instants \( k = \kappa_1, \kappa_2, \ldots, k_c \), instants \( \kappa_1 \) and \( k_c \) are under different constraint conditions. This means that a discontinuous change at instant \( \kappa_1 \) is detected at a later instant \( k_c \). The number, \( e_{\dot{\kappa}k} \), is a single scalar value and it is used to represent the difference of the models between the referential past instant and the current time instants, taking into account the uncertainties. A parameter estimate that is based on a time range including a discontinuity is not suitable. Therefore, when the discontinuities are detected, the estimation algorithm is reset by assigning all the elements of \( R_k, Q_k \) and \( F_k \) to 0 (or for numerical stability reasons very small such as \( 10^{-12} \)).

### 3.3.5. Batch Least Squares

Another common way of estimating the parameters of the model is using batch processing which can be performed offline. Batch processing differs from recursive identification in that it operates on a group of samples instead of each sample separately. According to this, the expression given by (3.13) can be used to constitute the following system of equations:

\[
\Phi_k^T = \Psi_k^T \Theta
\]

\[
\begin{bmatrix}
\phi_{i1}^T \\
\phi_{i2}^T \\
\vdots \\
\phi_{iN}^T
\end{bmatrix} =
\begin{bmatrix}
\psi_{i1}^T \\
\psi_{i2}^T \\
\vdots \\
\psi_{iN}^T
\end{bmatrix} \Theta 
\rightarrow 
\Phi = \Psi^\dagger \Theta
\]

(3.42)

(3.43)

From here, the least square estimate of the parameter matrix is,

\[
\Theta = \Psi^\dagger \Phi
\]

(3.44)

where \( \Psi^\dagger \) is the pseudoinverse of \( \Psi \), \( N \) is the number of samples used in the batch least
squares solution, $\Phi$ is the output matrix with $N \times 3$ dimensions, $\Psi$ is the regressor matrix with $N \times 10$ dimensions, and $\Theta$ is the parameter matrix with $10 \times 3$ dimensions. The batch processing can be applied separately for different segments of the task, since in different segments there will also be different parameters.
4. Impedance Control

Manipulation tasks (either domestic or industrial), such as the ones introduced in chapter 2 and 3, can be performed successfully and adequately by controlling the contact between the robot and the environment. Applying solely position control is not suitable for these tasks since obtaining a perfect model of both the environment (i.e. mechanical and geometrical properties of objects) and manipulator is not possible. Doing so may result in undesirably high contact forces, especially in the case of rigid contact conditions and can even cause damage either to the environment or to the robot. Therefore, the robot has to be capable of demonstrating a certain level of compliant behavior for the tasks involving contact. In this chapter, one commonly used interaction control method namely, impedance control, is introduced both for unconstrained (free) motion and constrained motion control phases of a manipulation task. Furthermore, the effect of joint elasticity and joint friction on this control law is also investigated. Unconstrained (free) motion is related to the case where no interaction with the environment is present, whereas this is not the case in constrained motion, where the manipulator is subject to kinematic constraints.

The manipulator that is used during both the simulations and experiments in this thesis, contains non-negligible joint elasticity (stiffness). Therefore it may be worthwhile to introduce this effect in the robot dynamical equations. The dynamic model of a flexible (elastic) joint robot together with joint friction is given as follows

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau + \tau_{\text{ext}} \]  \hspace{1cm} (4.1)

\[ B\dot{\theta} + \tau = \tau_m - \tau_f \]  \hspace{1cm} (4.2)

\[ \tau = K(\theta - q) \]  \hspace{1cm} (4.3)

Where \( n \) is the number of joints, \( q \in \mathbb{R}^n \) and \( \theta \in \mathbb{R}^n \), are the link and motor positions (reflected to link side of the gearbox, i.e. \( \theta = \frac{\theta_m}{i_g} \), where \( i_g \) is the transmission ratio) respectively. Furthermore, \( M(q) \in \mathbb{R}^{n \times n} \), is the link inertia matrix, \( C(q, \dot{q})\dot{q} \in \mathbb{R}^n \) represents the Coriolis and centrifugal terms, \( g(q) \in \mathbb{R}^n \) is the gravity torque vector, \( \tau \in \mathbb{R}^n \)'s are joint torques, \( \tau_f \in \mathbb{R}^n \)'s are the friction torques (such as Coulomb, Stibieck or LuGre type), and \( B = \text{diag}(b_i) \in \mathbb{R}^{n \times n} \) is the rotor inertia matrix (reflected to the link side of the gearbox) and \( K = \text{diag}(k_i) \in \mathbb{R}^{n \times n} \) is the joint stiffness matrix. The effect of external forces and moments acting on the manipulator are included to the model by \( \tau_{\text{ext}} \). In the case when the external forces and moments act on the end-effector, this can be given by

\[ \tau_{\text{ext}} = J^T(q)\left[ f_{\text{ext}} \quad n_{\text{ext}} \right]^T \]  \hspace{1cm} (4.4)

where \( f_{\text{ext}} \in \mathbb{R}^3 \) and \( n_{\text{ext}} \in \mathbb{R}^3 \) are external forces and moments respectively. Here, the end-effector forces and moments in the task space are mapped to the joint torques by using the manipulator Jacobian \( J(q) \). A damping term can also be included in (4.3), if the joint damping is not negligible:
\[ \tau = K(\theta - q) + D(\dot{\theta} - \dot{q}) \quad (4.5) \]

where \( D = \text{diag}(d_i) \in \mathbb{R}^{n \times n} \) represents the joint damping coefficient matrix [26, 27]. In this chapter, an impedance control approach that takes the joint flexibilities into account, as proposed in [25, 26, 27] is discussed. The control laws that are derived in this chapter make use of the notion of passivity; therefore, passivity is described here before the controller derivations. The stability and passivity properties of these controllers are only mentioned briefly in this chapter, further information can be obtained from the references. The passivity property is defined for a dynamical system given by the state-space model,

\[
\begin{align*}
\dot{x} &= f(x,u) \\
y &= h(x,u)
\end{align*} \quad (4.6)
\]

\[
\begin{align*}
\dot{x} &= f(x,u) \\
y &= h(x,u)
\end{align*} \quad (4.7)
\]

where \( f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n \) is locally Lipschitz, \( h : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p \) is continuous, with \( x \in \mathbb{R}^n \) being the state, \( u \in \mathbb{R}^m \) the input and \( y \in \mathbb{R}^n \) the output.

**Definition 4.1** The system (4.6) – (4.7) is said to be passive [31], if there exists a continuously differentiable positive semidefinite function \( V(x) \) (called the storage function) such that

\[
\begin{align*}
0 \leq u^T y - \dot{V} = \frac{\partial V}{\partial x} f(x,u), \\
\forall (x,u) \in \mathbb{R}^n \times \mathbb{R}^m
\end{align*} \quad (4.8)
\]

Other than this definition, some properties of the rigid body part of the model which will later on be used in stability proofs are mentioned in this section.

**Property 4.1** The link inertia matrix is symmetric and positive definite,

\[
M(q) = M(q) > 0, \quad \forall q \in \mathbb{R}^n.
\]

**Property 4.2** The matrix \( S(q, \dot{q}) = \dot{M}(q) - 2C(q, \dot{q}) \) is skew symmetric. This means that \( S + S^T = 0 \) and also \( z^T (\dot{M}(q) - 2C(q, \dot{q})) z = 0 \) for any \( z, q, \dot{q} \in \mathbb{R}^n \).

### 4.1. Free Motion Control

The execution of desired motions (trajectories) for a robot in the absence of any contact with the environment (i.e. \( \tau_{ext} = 0 \)) is called unconstrained (free) motion control. This can be achieved by applying position control either by means of joint variables or end-effector variables (i.e. Cartesian space). Robotic tasks such as spot welding, pick-and-place operations or spray-painting can be considered in this group. In this section both joint space and Cartesian space control laws are going to be introduced.

#### 4.1.1. Joint Space Control

Joint space control laws such as decentralized control (e.g. P.D., P.I.D, etc.) or centralized control (e.g. computed torque, adaptive control, etc.) aim at tracking a desired trajectory with
or regulating around a fixed setpoint given in the joint coordinates. A simplified diagram block diagram depicting the joint space control architecture is given below in Figure 4.1. Inverse kinematics is related to determining the joint coordinates (or variables), \( q \) for given end-effector position and orientation, \( x \). Thus it is the inverse of the mapping, \( x = A(q) \).

![Simplified diagram for joint space control](image)

The following controller structure in the joint space is discussed in this section.

\[
\begin{align*}
\tau_m &= B \cdot B_g \cdot u + \left( I - B \cdot B_g \right) \tau \\
u &= -K_g \left( \theta - \theta_{ref} \right) - D_g \left( \dot{\theta} - \dot{\theta}_{ref} \right) + B_g \ddot{\theta}_{ref} + \bar{g} (\theta)
\end{align*}
\]  \hspace{1cm} (4.9) \hspace{1cm} (4.10)

This control law contains a negative joint torque feedback by using a joint torque measurement (since \( B_g < B \)), a suitable gravity compensation term based on only motor position measurement that counterbalances the link side gravity torques in every stationary point and also velocity and acceleration feedforward terms to improve tracking performance. The joint torque feedback has two important features, first one is the reduction of the apparent motor (rotor) inertia, and the second one is the reduction of any motor side disturbance (e.g. friction torques \( \tau_f \)). The following closed loop system is obtained when the control law given by (4.9) – (4.10) is inserted in (4.1) – (4.2).

\[
\begin{align*}
M (q) \ddot{q} + C (q, \dot{q}) \dot{q} + g (q) &= \tau \\
B_g \left( \ddot{\theta} - \ddot{\theta}_{ref} \right) + D_g \left( \dot{\theta} - \dot{\theta}_{ref} \right) + K_g \left( \theta - \theta_{ref} \right) + \tau = \bar{g} (\theta) - B_g B^{-1} \tau_f
\end{align*}
\]  \hspace{1cm} (4.11) \hspace{1cm} (4.12)

It can be observed that the motor dynamics in closed-loop represented by (4.12) has its rotor inertia scaled down from \( B_g \) to \( B \), and the friction torques acting on the motor dynamics are also decreased by a factor of \( B_g B^{-1} \). The derivation of the gravity compensation term \( \bar{g} (\theta) \), and the related gravity potential function \( V_g (\theta) \) is only going to be mentioned very briefly here, further detailed information can be obtained from [25]. At steady-state (i.e. \( \theta = \theta_{eq} \), \( q = q_{eq} \)) the manipulator dynamics given by (4.1) become:

\[ g (q_{eq}) = K \left( \theta_{eq} - q_{eq} \right) \]  \hspace{1cm} (4.13)

which represents a holding torque between the torque due to joint flexibility and link side gravity terms. The equilibrium points for the motor angles can then be given as a function of the link angles by using (4.13)

\[ \theta_{eq} = q_{eq} + K^{-1} g (q_{eq}) \]  \hspace{1cm} (4.14)

which constitutes a set of stationary points \( \Omega = \left\{ (q_{eq}, \theta_{eq}) | \theta_{eq} = q_{eq} + K^{-1} g \left( q_{eq} \right) \right\} \). By using
(4.14) any motor position which is inside the set \( \Omega \) can also be expressed as a function of link side positions. Then by calculating the inverse of this function, namely

\[
q = h^{-1}(\theta) = \tilde{q}(\theta)
\]  

(4.15)

the gravity compensation term \( g(q) = g(\tilde{q}(\theta)) = \bar{g}(\theta) \), can be calculated as a function of motor positions. The gravity potential function \( V_g(\theta) \) based on motor angles, has the following property

\[
\frac{\partial V_g(\theta)}{\partial \theta} = \frac{\partial V_g(\tilde{q})}{\partial \tilde{q}} = g(\tilde{q}(\theta))^T = \bar{g}(\theta)^T
\] 

(4.16)

The passivity and the stability of the closed loop system given by (4.11) – (4.12) is going to be derived for the case of a constant setpoint (i.e. \( \dot{\theta}_{ref} = 0 \), \( \ddot{\theta}_{ref} = 0 \)). The passivity of (4.11), related to the mapping \( \tau \rightarrow \dot{q} \), can be shown by using the following storage function \( S_q \),

\[
S_q = \frac{1}{2} \dot{q}^T M(q) \dot{q} + V_g(q)
\] 

(4.17)

where the first term represents the kinetic energy of the rigid body part of the manipulator model and the second term represents the potential energy due to gravity. When (4.17) is differentiated and the properties given at the beginning of this chapter are used,

\[
\dot{S}_q = \frac{1}{2} \left[ \dot{q}^T M(q) \dot{q} + \dot{\dot{q}}^T \left( \dot{M}(q) \dot{q} + M(q) \ddot{q} \right) \right] + \dot{V}_g(q)
\]

\[
\dot{S}_q = \frac{1}{2} \left[ 2 \dot{q}^T M(q) \dot{q} + \dot{\dot{q}}^T M(q) \dot{q} \right] + \dot{q}^T g(q)
\]

\[
\dot{S}_q = \frac{1}{2} \dot{q}^T \left[ 2(\tau - C(q, \dot{q}) \dot{q} - g(q)) + \dot{M}(q) \dot{q} + 2g(q) \right]
\]

\[
\dot{S}_q = \dot{\dot{q}}^T \tau
\] 

(4.18)

Therefore (4.18) satisfies the definition (4.8). The passivity of (4.12), related to the mapping \( \dot{q} \rightarrow -\tau \), can be presented with the storage function \( S_\theta \),

\[
S_\theta = \frac{1}{2} \dot{\theta}^T B_\theta \dot{\theta} + \frac{1}{2} (\theta - q)^T K(\theta - q) + \frac{1}{2} \dot{\theta}^T K_\theta \dot{\theta} - V_\pi(\theta)
\] 

(4.19)

where the first term represents the kinetic energy of the rotor dynamics, the second term represents the potential energy stored in the joint flexibilities and the last two terms are the potential energy stored in the controller and gravity potential respectively. When (4.19) is differentiated and (4.12) is used, we obtain

\[
\dot{S}_\theta = \dot{\theta}^T B_\theta \dot{\theta} + (\dot{\theta} - \dot{q})^T K(\theta - q) + \dot{\theta}^T K_\theta \dot{\theta} - \dot{V}_\pi(\theta)
\]

\[
\dot{S}_\theta = \dot{\theta}^T \left[ \bar{g}(\theta) - K_\theta \dot{\theta} - D_\theta \dot{\theta} - \tau \right] + (\dot{\theta} - \dot{q})^T \tau + \dot{\theta}^T K_\theta \dot{\theta} - \dot{\theta}^T \bar{g}(\theta)
\]
\[ \dot{S}_q = -\dot{\theta}^T D_\theta \dot{\theta} - \dot{q}^T \tau \leq -\dot{q}^T \tau \]  

(4.20)

Therefore (4.20) satisfies the passivity property given by definition (4.8). The overall closed loop system, (4.11) – (4.12) can be represented as feedback interconnection of the two subsystems, namely the link side and the motor side dynamics, by using their derived passivity properties. The feedback interconnection of these two passive subsystems is displayed in Figure 4.2.

The stability of this closed loop system can be shown with the following candidate Lyapunov function

\[ V(q, \dot{q}, \theta, \dot{\theta}) = S_q + S_\theta \]

(4.21)

The derivation for the positive definiteness of this function can be found in [25]. The derivative of (4.21) along the solutions of the closed loop system

\[ \dot{V} = \dot{S}_q + \dot{S}_\theta = -\dot{\theta}^T D_\theta \dot{\theta} \]

(4.22)

is negative semi-definite since \( D_\theta \) is a positive definite matrix, which shows that the equilibrium point is stable. In order to show that it is asymptotically stable LaSalle’s invariance principle can be used. From the closed-loop dynamics (4.11) – (4.12) and (4.22) it can be observed that there does not exist any trajectory for which \( \theta = 0 \) except the equilibrium \( \theta = \theta_{eq}, q = q_{eq} \). Asymptotic stability follows from here.

4.1.2. Cartesian Space Control

As opposed to the joint space control laws, Cartesian space control laws operate directly on end-effector variables. In the case that the end-effector variables cannot be measured directly, they can be calculated by using the forward kinematics mapping, from \( q \) to \( x \), i.e. \( x = \Lambda(q) \).

A simplified diagram block diagram depicting the Cartesian space control architecture is given below in Figure 4.3.

\[ X_{\text{desired}} \]  

\[ \text{Controller} \]  

\[ \text{Robot} \]  

\[ x \]
The following control law in the Cartesian space is discussed in this section.

\[
\begin{align*}
\tau_{m} &= B \cdot B_{o}^{-1} u + \left(I - B \cdot B_{o}^{-1}\right) \tau \\
u &= -J(\theta)^{T} \left(K_{x}(\Lambda(\theta) - x_{ref}) + D_{x}(\dot{x}(\theta) - \dot{x}_{ref})\right) + B_{o} \ddot{\theta}_{ref} + \bar{g}(\theta) \quad (4.23)
\end{align*}
\]

where \(\dot{x}(\theta) = \Lambda(\theta) - x_{ref}\) is the Cartesian error, \(x_{ref}\) represents the Cartesian reference signal, and \(\dot{x}(\theta) = J(\theta) \dot{\theta}\) represents the end-effector velocity in Cartesian coordinates. Moreover, \(K_{x}\) is the Cartesian stiffness matrix and \(D_{x}\) is the Cartesian damping matrix. The following closed loop system is obtained when the control law given by (4.23) – (4.24) is inserted in (4.1) – (4.2),

\[
\begin{align*}
M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) &= \tau \\
B_{o}(\ddot{\theta} - \ddot{\theta}_{ref}) + J(\theta)^{T}\left[K_{x}(\Lambda(\theta) - x_{ref}) + D_{x}(\dot{x}(\theta) - \dot{x}_{ref})\right] + \tau = \bar{g}(\theta) - B_{o}B_{o}^{-1}\tau_{f} \quad (4.25)
\end{align*}
\]

It can be observed from (4.25) – (4.26) that the effects of the joint torque feedback mentioned for the joint space controller still remain in this Cartesian control law. The passivity and stability analysis of this control law can be found in the Appendix A. The stiffness in the Cartesian control law given by (4.23) – (4.24) can also be expressed as a configuration dependent joint stiffness matrix if it is desired to drive the manipulator by using joint space coordinates [36]. By using the relation between the virtual joint displacements and virtual end-effector displacements

\[
\delta x = J(\theta) \delta \theta \quad (4.27)
\]

the error in Cartesian coordinates can be rewritten in joint coordinates as,

\[
\begin{align*}
\ddot{x}(\theta) &\approx J(\theta) \dot{\theta} \\
\Lambda(\theta) - x_{ref} &\approx J(\theta)(\theta - \theta_{ref}) \quad (4.28)
\end{align*}
\]

Then by using (4.28), the joint space approximation of this Cartesian control law, (4.24) can be given as,

\[
\begin{align*}
u &= -J(\theta)^{T}\left[K_{x}(\Lambda(\theta) - x_{ref}) + D_{x}(\dot{x}(\theta) - \dot{x}_{ref})\right] + B_{o} \ddot{\theta}_{ref} + \bar{g}(\theta) \\
u &= -J(\theta)^{T}\left[K_{x}J(\theta)(\theta - \theta_{ref}) + D_{x}J(\theta)(\dot{\theta} - \dot{\theta}_{ref})\right] + B_{o} \ddot{\theta}_{ref} + \bar{g}(\theta) \quad (4.29)
\end{align*}
\]

Since the term \(D_{x}J(\theta)\) in (4.29) is used to improve the system damping with the control law, it can also be rewritten in joint space as

\[
u = -J(\theta)^{T}K_{x}J(\theta)(\theta - \theta_{ref}) - D_{o}(\dot{\theta} - \dot{\theta}_{ref}) + B_{o} \ddot{\theta}_{ref} + \bar{g}(\theta) \quad (4.30)
\]
4.2. **Constrained Motion Control**

Constrained motion control, as mentioned earlier, is related to the case when the robot is in contact with the environment. The Cartesian control law given by (4.23) – (4.24) is going to be treated for the constrained motion control. A simplified environment model, which is characterized by only stiffness forces, is utilized for the analysis given in this section. In addition, the steady-state behavior of the system with the suggested controller is also derived [29]. The steady-state derivation is first performed in the case when a diagonal stiffness matrix with equal stiffnesses in each direction is selected, then it is extended to the case with different stiffness selections for unconstrained/constrained directions. The effect of contact friction (i.e. Coulomb friction) that occurs when moving on a surface, is neglected in the analysis introduced in this section. Furthermore, no contact moments are taken into account, i.e. a task with only translational degrees of freedom is considered. This task involves applying a force normal to a surface and following a trajectory in the tangential directions of the surface (such as writing on a blackboard, or cleaning a table, etc.). The reaction force that the environment exerts on the manipulator, based on the simplified environment assumption is expressed as,

\[
f_{ext} = -K_E (x - x_E)
\]  

(4.31)

where \(x_E\) is the point of restitution (i.e. the location to which the contact point returns when the end-effector is not in contact with the environment) and \(K_E\) is a positive semidefinite matrix. The environment stiffness matrix is given by

\[
K_E = k_E nn^T
\]  

(4.32)

where \(k_E\) is the scalar stiffness coefficient of the environment and \(n \in \mathbb{R}^3\) represents the contact normal (i.e. the constrained direction).

**Property 4.3** The constrained and unconstrained directions have the following mathematical properties:

1. \((nn^T)(nn^T) = nn^T\)
2. \((nn^T)(I - nn^T) = nn^T - nn^T nn^T = 0_{3x3}\)

where \(I \in \mathbb{R}^{3x3}\) is the identity matrix.

The closed-loop dynamics using the Cartesian control law together with the simple environment model becomes

\[
M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) = \tau - J^T(q) k_E nn^T [x(q) - x_E]
\]  

(4.33)

\[
B_\theta \ddot{\theta} + J(\theta)^T [K_c \ddot{x}(\theta) + D_c \dot{x}(\theta)] + \tau = \dot{\gamma}(\theta) - B_\theta \dot{J}^T \tau_f
\]  

(4.34)

At steady state, this equation simplifies to
\[ g(q_{eq}) = K(\theta_{eq} - q_{eq}) - J^T(q_{eq})k_Enn^T[x(q_{eq}) - x_E] \]  
(4.35)

\[ J(\theta_{eq})^T K_x x(\theta_{eq}) - x_{ref} + K(\theta_{eq} - q_{eq}) = \bar{g}(\theta_{eq}) - B_oB^{-1}\tau_{f,eq} \]  
(4.36)

Hereafter it will be assumed that at steady state for small displacements \( J(q_{eq}) \approx J(\theta_{eq}) \), \( x(q_{eq}) \approx x(\theta_{eq}) = x_E \) and \( g(q_{eq}) \approx \bar{g}(\theta_{eq}) \). An exact Cartesian stiffness design method that can overcome the limitations of this assumption is given in [27]. Combining (4.35) and (4.36)

\[ J(\theta_{eq})^T K_x(x_{eq} - x_{ref}) = J^T(\theta_{eq})k_Enn^T[x_E - x_{eq}] - B_oB^{-1}\tau_{f,eq} \]  
(4.37)

when (4.37) is rearranged,

\[ \delta_{eq} = k_Enn^T(x_E - x_{eq}) - K_x(x_{eq} - x_{ref}) \]

\[ 0_{3x1} = k_Enn^Tx_E - k_xnn^Tx_{eq} - K_xx_{eq} + K_x x_{ref} - \delta_{eq} \]  
(4.38)

where \( \delta_{eq} = J(\theta_{eq})^{-T}B_oB^{-1}\tau_{f,eq} \) represents the steady-state values of the remaining friction torques (e.g. static friction). The steady-state value of the reaction force which the robot exerts on the environment can be obtained as follows,

\[ f_{ext,eq} = k_Enn^T(x_{eq} - x_E) \]  
(4.39)

The steady state analysis is first going to be performed in the case when the same stiffness values are selected for all directions (i.e. both constrained and unconstrained directions, \( K_x = k_xI \), where \( k_x \) is a scalar). The steady state value of the end-effector position in the constrained directions can be calculated by using the property 4.3. When both sides of (4.38) are multiplied by \( nn^T \),

\[ nn^T \cdot 0_{3x1} = k_Enn^Tnn^Tx_E - k_xnn^Tx_{eq} + k_xnn^Tx_{ref} - nn^T\delta_{eq} \]

\[ 0_{3x1} = nn^T[k_E x_E - k_x x_{eq} + k_x x_{ref} - \delta_{eq}] \]  
(4.40)

Since \( nn^T \neq 0_{3x1} \), the terms inside the brackets need to equal zero,

\[ 0_{3x1} = k_E x_E - k_x x_{eq} + k_x x_{ref} - \delta_{eq} \]

\[ (k_E + k_x) x_{eq} = k_x x_{ref} + k_E x_E - \delta_{eq} \]

\[ x_{eq} = \frac{k_x}{k_E + k_x} \left( x_{ref} - \frac{1}{k_x} \delta_{eq} \right) + \frac{k_E}{k_E + k_x} x_E \]  
(4.41)

The steady state value of the end-effector position in the unconstrained directions can be calculated by multiplying both sides of (4.38), by \( I - nn^T \),

\[ (I - nn^T) \cdot 0_{3x1} = k_E(I - nn^T)nn^T(x_E - x_{eq}) + k_x(I - nn^T)(x_{ref} - x_{eq}) - (I - nn^T)\delta_{eq} \]
Since \( I - nn^T \neq 0_{3 \times 1} \), the terms inside the brackets are zero,

\[
0_{3 \times 1} = k_x x_{ref} - k_x x_{eq} - \delta_e
\]

\[
x_{eq} = x_{ref} - \frac{1}{k_x} \delta_e
\]

Therefore the end-effector position in all directions can be given by combining (4.41) and (4.43),

\[
x_{eq} = nn^T \left[ \frac{k_x}{k_E + k_x} \left( x_{ref} - \frac{1}{k_x} \delta_e \right) + \frac{k_E}{k_E + k_x} x_E - x_E \right] + \left( I - nn^T \right) \left[ x_{ref} - \frac{1}{k_x} \delta_e \right]
\]

The steady state value of the force that the manipulator applies on the environment is given by (4.39), (4.41)

\[
f_{eq,\text{ext,xx}} = k_E nn^T \left[ \frac{k_x}{k_E + k_x} \left( x_{ref} - \frac{1}{k_x} \delta_e \right) + \frac{k_E}{k_E + k_x} x_E - x_E \right]
\]

\[
f_{eq,\text{ext,yy}} = k_E nn^T \left[ \frac{k_x}{k_E + k_x} \left( x_{ref} - \frac{1}{k_x} \delta_e \right) + \frac{k_E}{k_E + k_x} x_E - x_E \right]
\]

\[
f_{eq,\text{ext,zz}} = \frac{k_x k_E}{k_E + k_x} nn^T \left[ x_{ref} - \frac{1}{k_x} \delta_e - x_E \right]
\]

This means that for very stiff environments \( k_E > > k_x \), the equivalent stiffness can roughly be approximated by \( \frac{k_x k_E}{k_E + k_x} \approx k_x \) and for very compliant environments \( k_x > > k_E \), the equivalent stiffness can roughly be approximated by \( \frac{k_x k_E}{k_E + k_x} \approx k_E \). This means that in very stiff environments selecting the controller stiffness \( k_x \) high, can improve reducing the steady state effects of friction which can be noticed from (4.44) and (4.45), however it also causes the contact forces to increase. Therefore, instead of selecting the stiffness matrix as a diagonal matrix with the same entries, the stiffness matrix can also be selected as \( K_x = k_f nn^T + k_p \left( I - nn^T \right) \), where \( k_f \) represents the low stiffness values in constrained and \( k_p \) represents the high stiffness values in unconstrained directions. The steady state value of the end-effector position in the constrained directions can be calculated by using the property 4.3. When both sides of (4.38) are multiplied by \( nn^T \),

\[
nn^T \cdot 0_{3 \times 1} = nn^T \left[ k_x x_E - k_x x_{eq} \right] - nn^T \left[ k_f nn^T + k_p \left( I - nn^T \right) \right] \left( x_{eq} - x_{ref} \right) - nn^T \delta_e
\]
\[ 0_{3\times1} = k_{E}nn^{T}x_{E} - k_{E}nn^{T}x_{eq} - k_{f}nn^{T}(x_{eq} - x_{ref}) - nn^{T}\delta_{eq} \]
\[ 0_{3\times1} = nn^{T}[k_{E}x_{E} - k_{E}x_{eq} - k_{f}x_{eq} + k_{f}x_{ref} - \delta_{eq}] \] (4.46)

Since \( nn^{T} \neq 0_{3\times1} \), the terms inside the brackets are zero,
\[ x_{eq} = \frac{k_{f}}{k_{E} + k_{f}}(x_{ref} - \frac{1}{k_{f}}\delta_{eq}) + \frac{k_{E}}{k_{E} + k_{f}}\delta_{eq} \] (4.47)

The steady state value of the end-effector position in the unconstrained directions can be calculated by multiplying both sides of (4.38), by \( I - nn^{T} \),
\[ (I - nn^{T}) \cdot 0_{3\times1} = (I - nn^{T})[k_{E}nn^{T}(x_{E} - x_{eq}) - (k_{f}nn^{T} + k_{p}(I - nn^{T}))(x_{eq} - x_{ref}) - \delta_{eq}] \]
\[ 0_{3\times1} = (I - nn^{T})[-k_{p}(x_{eq} - x_{ref}) - \delta_{eq}] \] (4.48)

Since \( I - nn^{T} \neq 0_{3\times1} \), the terms inside the brackets are zero,
\[ x_{eq} = x_{ref} - \frac{1}{k_{p}}\delta_{eq} \] (4.49)

Therefore the end-effector position in all directions is given by combining (4.47) and (4.49),
\[ x_{eq} = nn^{T}\left[\frac{k_{f}}{k_{E} + k_{f}}(x_{ref} - \frac{1}{k_{f}}\delta_{eq}) + \frac{k_{E}}{k_{E} + k_{f}}x_{E}\right] + (I - nn^{T})\left[x_{ref} - \frac{1}{k_{p}}\delta_{eq}\right] \] (4.50)

The steady state value of the force that the manipulator applies on the environment can be given by using (4.39), (4.47)
\[ f_{ext,eq} = k_{E}nn^{T}\left[\frac{k_{f}}{k_{E} + k_{f}}(x_{ref} - \frac{1}{k_{f}}\delta_{eq}) + \frac{k_{E}}{k_{E} + k_{f}}x_{E} - x_{eq}\right] \]
\[ f_{ext,eq} = k_{E}nn^{T}\left[\frac{k_{f}}{k_{E} + k_{f}}(x_{ref} - \frac{1}{k_{f}}\delta_{eq}) + k_{f}x_{E} - k_{f}x_{E} - k_{f}x_{eq}\right] \]
\[ f_{ext,eq} = k_{E}nn^{T}\left[\frac{k_{f}}{k_{E} + k_{f}}(x_{ref} - \frac{1}{k_{f}}\delta_{eq}) - x_{E}\right] \] (4.51)

From this comparison between (4.45) and (4.51), it can be observed that different stiffnesses can be selected for constrained and unconstrained directions. Furthermore, selecting low stiffness (or equivalently high compliance) in the constrained directions leads to a lower interaction force at the steady state. Therefore this strategy can be used to control interaction forces indirectly and furthermore limit them to an acceptable value that should stress neither
the interacted object nor the robot. The passivity and stability analysis of this control law can be found in the Appendix A.

### 4.3. Contact Transition Control

An important control issue which arises during a manipulation task is the stable execution of the transition phase from free motion to constrained motion. This issue becomes more evident when the manipulator makes hard contact with the environment. Hard contact is an impact that happens when the robot contacts a stiff environment with a high velocity. Different contact transition controller designs can be found in [24, 28, 30].

The primary prerequisite for either changing the control law from position to force control, or varying the impedance of an impedance controller is the ability to detect contact with the environment. There are different ways to detect contact with a surface, either by using position measurements (e.g. range sensors, cameras) or by using force measurements (e.g. force/torque sensors). Since obtaining a very precise measurement of the location of the contacted environment by using position measurements is practically not feasible, transition is made based on the force/torque measurements. A common encountered problem which occurs during the transition phase is called “chattering”. It causes the end-effector of the robot to bounce continuously on the contacted environment without being able to establish contact and exert forces on the environment, thus creating instability. As mentioned before, contact with a surface can be detected by using force measurements and this can also be used to switch between free and constrained motion control. However, instead of instantaneously switching between the two control laws, it might be better to introduce some sort of a safety region for the switching action. This safety region can help preventing the end-effector to bounce off the surface of the environment when it first establishes contact. Hysteresis action is a mechanism which can introduce such a safety region, and it can be used on the measured force sensor signals [24]. The switching strategy when hysteresis used between the two different phases (free/constrained) is depicted below in Figure 4.4.

![Figure 4.4: Contact transition method by using hysteresis](image)

In this figure, the variable Q represents a logical variable (i.e. a switch) to determine when it is suitable to switch the control law from position to force control or vary the gains of the impedance controller. The switching following the path 0 → I → II → III occurs when the contact force \( f_{contact} \) exceeds \( f_{thres,high} \). The switching following the path III → IV → V → 0 occurs when the contact force \( f_{contact} \) becomes lower than \( f_{thres,low} \). Therefore, some sort of a safety region is provided for the switching mechanism. Further information, stability proofs and more sophisticated versions of this approach can be found in [24].
4.4. Summary

In this chapter an impedance control law which can be used to perform manipulation tasks has been introduced. The influence of joint flexibility and friction on the control law has been discussed. The concept of passivity has been introduced since the impedance controller makes use of this notion. The application of the approach to free motion and constrained motion cases has been discussed. The implementation of the impedance controller in both joint and Cartesian coordinates has been introduced. The stability proof and passivity derivation for the joint space impedance controller has been given. Furthermore, a joint space approximation for the Cartesian space control law is also introduced. A steady-state analysis of the Cartesian impedance controller has been derived. The influence of selecting different stiffness in different task frame directions has been shown. Finally a method which can handle transitions between free and constrained motion phases has been discussed.
5. Simulation Results

In this chapter, first a brief introduction of the simulation model is presented, and then the results of the contact impedance identification simulations based on the method explained in chapter 3 are introduced. Finally, the results obtained by using the impedance controller described in chapter 4 are presented.

Since the available experimental setup contains only 2 degrees-of-freedom, the simulations are based on a 2 d.o.f. model (4.1) – (4.3). The details and the derivation of the dynamical equations for this simulation model can be found in the Appendix B. The inertia of the motor reflected to the link side, that is used during the simulations, is taken equal to that of the experiments (i.e. $B = 0.1592 \text{ kg*m}^2$). Motor side friction and joint damping has not been modeled in the simulations. A schematic drawing of the manipulator while it is in contact with the simulated environment is presented in Figure 5.1. The directions of the base frame (i.e. $X_0-Y_0-Z_0$) together with the positive directions of the joint angles (i.e. $q_1-q_2$) are also indicated in this figure. The positive $X_0$ direction used in simulations can be compared with the negative $X$ direction used in the experiments shown in Figure 6.1, the positive $Y_0$ direction can be compared with the negative $Z$ direction in the experiments, and the positive $Z_0$ direction can be compared with the negative $Y$ direction in the experiments.

The simulations are performed with Matlab/Simulink, by using the ode5 solver with a time-step size of 0.001 seconds.

5.1. Identification of contact impedance

In this section simulation results which are related to identifying the contact impedance are presented. The arm has two rotational degrees-of-freedom, thus it is not possible to independently excite the environment in all three directions. Therefore one of the directions, namely the z-direction, is selected as the inherent motion direction and impedance perception.
is not performed in that direction. The simulation is related to sliding on a planar object such as the one shown in Figure 5.1. The contacted environment which is used in simulations is modeled as a combination of a linear spring and damper. The parameters which are used during the simulations are presented in Table 5.1.

<table>
<thead>
<tr>
<th>Table 5.1 Table of simulation parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact stiffness [N/m]</td>
</tr>
<tr>
<td>Contact damping [Ns/m]</td>
</tr>
<tr>
<td>Env. y-coordinate [m]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Controller stiffness [N/m]</th>
<th>Unconstrained motion</th>
<th>Constrained motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller damping [Ns/m]</td>
<td>$D_{xx} = 400$</td>
<td>$D_{vy} = 400$</td>
</tr>
<tr>
<td>Inertia scaling [kg*m²]</td>
<td>$B_{\theta} = 0.1592$</td>
<td>$B_{\theta} = 0.1592$</td>
</tr>
</tbody>
</table>

The simulation scenario consists of three phases:

1) **Unconstrained motion**: Driving the arm from its initial posture through free space until its tip makes a contact with a horizontally placed object (which is parallel to the $X_0-Z_0$ plane indicated in Figure 5.1).

2) **Constrained motion**: Making a wiping motion over the surface of the object while at the same time applying a force on it, and finally,

3) **Unconstrained motion**: Driving the arm back to its initial posture.

In constrained motion phase, the controller stiffness in the $y$-direction is selected much smaller compared to the stiffness in the $x$-direction, since $y$-direction is the constrained direction and forces need to be controlled in that direction. During the transition from free to constrained motion and vice versa, the strategy explained in section 4.3 is used. According to this strategy, the upper bound of the hysteresis on the force signals is selected as $f_{\text{hres, high}} = 2.5$ N, whereas the lower bound is selected as $f_{\text{hres, low}} = 0$ N. The gains are interpolated by smoothing the stepwise changes using the following 2nd order filter,

$$ G(s) = \frac{\omega_n^2}{s^2 + 2\zeta \omega_n + \omega_n^2} \quad (5.1) $$

where $\omega_n$ is used for determining the speed of gain change and $\zeta$ determines the damping of the filter. In the simulations these are selected as $\omega_n = 30 \text{ rad/s}$ and $\zeta = 1$. The end-effector force in $y$-direction recorded during the simulations can be observed in Figure 5.2. It can be observed that when the tip is pressed down the force in the $y$-direction displays a significant change. The forces in the other directions (i.e. $x$ and $z$) are zero, since no stiffness or contact friction has not been modeled in those directions in the simulations. The end-effector positions are displayed in Figure 5.3. The straight lines represent the recorded end-effector positions, and the dashed lines represent the reference trajectories designed to execute the simulated scenario.
The parameters of the virtual soft finger (V.S.F.) which are used during the simulation are selected as $K_v = 700 \text{ N/m}$ and $B_v = 10 \text{ Ns/m}$. The parameters of the forgetting factor are selected as $X_H = 0.02 \text{ m}$ and $T_H = 1 \text{ sec}$. The other parameters that are used during the simulations are $A = 0.1$ for the uncertainty estimation parameter and $U = 0.1$ for the discontinuity detection parameter. The norm of the end-effector velocity is given in Figure 5.4. Since only the x and y components of both force and position data have been used, the norm of the velocity is calculated from differentiating the x and y position data after the virtual soft finger algorithm is applied. According to the selection of the forgetting factor parameters, the bound on the low-speed motion becomes $X_H / T_H = 0.02 \text{ m/s}$.
Figure 5.4 Norm of the end-effector velocity

The results of the impedance perception algorithm are given in Figures 5.5, 5.6 and 5.7. The straight lines in these figures represent estimated impedance components by the recursive algorithm, the horizontal dash-dot lines are the estimated values by using the batch least squares algorithm, the gray bands represent the uncertainties of the estimated impedance components, and the vertical dotted lines indicated in Figure 5.5 represent the instants when the discontinuities are detected (i.e. gain and loss of contact for this case). The batch processing is applied when the end-effector is in constrained motion phase. The bias terms which are estimated by the impedance perception algorithm are given in Figure 5.5. The estimated discontinuity instants are close to the real instants when the discontinuities occur, which can also be seen from the end-effector force plot in Figure 5.2, since at around 1.96 seconds the end-effector force in y-direction become different from zero and start increasing, while around 8.03 seconds it becomes zero again.

Figure 5.5 Estimated bias terms by recursive/batch algorithms, uncertainties, and discontinuities
Here the discontinuities are detected by using the second method based on the difference between past and present model outputs introduced in section 3.3.4. The stiffness terms which are obtained by the impedance estimation algorithm are given in Figure 5.6.

![Figure 5.6 Estimated stiffness terms by recursive/batch algorithms and corresponding uncertainties](image)

The apparent stiffness of the environment in combination with the V.S.F.’s stiffness can be calculated as,

$$k_{E} \cdot K_{v} = \frac{5 \cdot 10^4 \cdot 700}{5 \cdot 10^4 + 700} \approx 690.35 \text{ [N/m]}$$

which is close to the virtual soft finger’s (V.S.F.) stiffness. It can be observed that the stiffness value $K_{yy}$, when the end-effector is in contact with the surface, converges very close to the apparent stiffness, with a small uncertainty. This is because the surface normal is in the y-direction and the environment is stiff. It can also be observed that the off diagonal term, $K_{xy}$ converge very closely to the correct zero value. The stiffness in the tangential direction $K_{xx}$, and the other off diagonal term, $K_{yx}$ when the end-effector is in contact with the surface, which are not shown here, are both zero. There are also some transient effects because of the resetting of the algorithm when the phase changes from unconstrained to constrained motion and vice versa. These transient effects vanish approximately after 0.5 second. Both the bias and the stiffness values in both of the unconstrained motion phases are close to zero, with a low uncertainty when the short initial transient effects are excluded. This can of course be expected, since there is no reaction force coming from the environment in these segments. The consistency of the bias and stiffness terms in y-direction can also be checked by calculating the absolute value of the flat surface’s y coordinate by using the values obtained from batch processing in constrained motion phase as follows

$$c_y = \left| \frac{-343.9}{694.7} \right| \approx 0.495 \text{[m]}$$

which is very close to the real coordinate of the surface. The damping terms that are
obtained by the impedance estimation algorithm are given in Figure 5.7. It can be seen that the damping $B_{yy}$ is close to $B_y$ with a small uncertainty, since the surface normal is in $y$-direction. It can be observed that in the constrained motion phase, the damping $B_{xy}$ is very close to zero with a small uncertainty. The damping in the tangential direction $B_{xx}$ and the other off diagonal damping $B_{yx}$, which are not shown here are both zero. There are also some transient effects because of the resetting of the algorithm, however, these effects vanish approximately after 0.5 second.

The related mass matrix terms that are obtained by the impedance estimation algorithm are given in the Appendix F. The estimator’s performance can also be checked by comparing the outputs (i.e. forces) of the model given by,

$$\hat{\phi}_k = \Theta_k^T \Psi_k$$  \hspace{2cm} (5.2)

with the simulated (raw) outputs. The output is estimated using the following relationship:

$$\hat{\phi}_k = \hat{\Theta}_k^T \Psi_k$$  \hspace{2cm} (5.3)

The 2nd element of the simulated and the estimated outputs and their difference are given in Figure 5.8. The estimated output and the estimation error for the 1st element are both zero. It can be observed that the estimated outputs are matching the sensed signal sufficiently well, except when discontinuities occur. The effect of these discontinuities in the estimated impedance parameters and in the output are present only for a short period of time.
5.2. Results of Control Simulations

5.2.1. Unconstrained Motion Control

In this section, the simulation results related to free motion control are given. First the results related to joint space impedance controller given by

\[ \tau_m = B \cdot B_\theta^{-1} u + \left( I - B \cdot B_\theta^{-1} \right) \tau \]

\[ u = -K_\theta \left( \theta - \theta_{ref} \right) - D_\theta \left( \dot{\theta} - \dot{\theta}_{ref} \right) + B_\theta \ddot{\theta}_{ref} + \bar{g}(\theta) \]

are presented, then the results related to Cartesian space impedance controller

\[ \tau_m = B \cdot B_\theta^{-1} u + \left( I - B \cdot B_\theta^{-1} \right) \tau \]

\[ u = -J(\theta)^	op \left( K_x \left( \theta - x_{ref} \right) + D_x \left( \dot{x}(\theta) - \dot{x}_{ref} \right) \right) + B_x \ddot{x}_{ref} + \bar{g}(\theta) \]

and its joint space approximation

\[ \tau_m = B \cdot B_\theta^{-1} u + \left( I - B \cdot B_\theta^{-1} \right) \tau \]

\[ u = -J(\theta)^	op K_x J(\theta) \left( \theta - \theta_{ref} \right) - D_\theta \left( \dot{\theta} - \dot{\theta}_{ref} \right) + B_\theta \ddot{\theta}_{ref} + \bar{g}(\theta) \]

are given. As mentioned at the beginning of this chapter, the model which is used during in the simulations has only 2 degrees-of-freedom. This means that the workspace of the robot is a nearly spherical 2D surface in the Cartesian space. Therefore, a special attention is given for the design of the trajectories and control in the Cartesian space case. There are two possibilities: 1) we can specify the desired trajectories along two Cartesian directions (e.g. x and y) and calculate the third one by using the kinematic equations of the robot; motion control is performed in all three directions, 2) we specify the desired trajectories in two
directions and use motion in these two directions only. The modified gravity compensation in both control laws contains the inverse function calculation (4.15). Since gravity torques in general contain a combination of trigonometric functions, the analytical solution of such inverse functions is very difficult or sometimes even not possible. However, this inverse function can be calculated using numerical methods or by neural networks. Here, the numerical procedure suggested in [25] is used. This numerical procedure can be derived by rearranging (4.14) as

\[ T(q) = \theta - K^{-1}g(q) \]  

(5.10)

This function can then be solved iteratively as,

\[ \hat{q}_{n+1} = T(\hat{q}_n) \]  

(5.11)

where \( n \) is the iteration number. This iteration is performed in every time step of the controller implementation. The initial value for this iteration loop is selected as \( \hat{q}_0 = \theta \), which is the motor side position in each control cycle. More details about this iterative procedure and its convergence properties can be found in [25]. In all simulations, \( n = 5 \) iterations provides satisfactory tracking errors.

A setpoint regulation simulation is conducted using the joint space impedance control law, in order to show that the link-side position errors converge to zero, with the modification (4.14) – (4.15) and the numerical routine (5.10) – (5.11) when the robot executes free motion. The robot is commanded to make a step-wise movement of \( \pi/4 \) rad in the 1st joint and -\( \pi/4 \) in the 2nd joint. The results of this simulation are depicted in Figure 5.9. The straight lines on the left are related to the link-side positions and the dash-dot lines are the setpoint signals, the difference between these two signals are plotted on the right-hand side of the figure. The controller parameters which are used during this simulation are \( K_\theta = 900 \text{ Nm/rad} \) for the controller stiffness, \( D_\theta = 60 \text{ Nms/rad} \) for the controller damping, and \( B_\theta = 0.1592 \text{ kg*m}^2 \) for inertia scaling (i.e. no negative joint torque feedback).

![Figure 5.9](image)

Figure 5.9 Results for the setpoint regulation using the joint impedance controller without the joint torque feedback.
It can be observed from Figure 5.9 that the link-side position errors converge to zero at steady-state when the aforementioned modification for the gravity compensation is used.

In the second simulation consecutive point to point movements are made using the Cartesian space impedance control law. The point to point trajectories for the Cartesian coordinates can be generated in many ways, such as by means of cubic or, quintic polynomials, splines, etc [37]. In this work, these trajectories are obtained by smoothing the step inputs with the following 4th order filter,

\[
G(s) = \left(\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right)^2
\]

(5.4)

where \(\omega_n\) is used for determining the speed of the movement and \(\zeta\) determines the damping of the filter. The reference velocities and accelerations can be obtained either by differentiating the output of this filter or by using the outputs of the integrators used in this filter. The implementation of this point-to-point trajectory generator can be found in the Appendix. The position, velocity and acceleration profiles that correspond to such a reference with \(\omega_n = 4\) rad/s, \(\zeta = 1\), are presented in Figure 5.10.

The results of this simulation are presented in Figure 5.11. The straight lines are related to the link-side positions and the dash-dot lines are the reference trajectories, the error between these two signals are given in Figure 5.11. The controller parameters in this simulation are \(K_{xx} = K_{yy} = K_{zz} = 35000\) N/m for the controller stiffness, \(D_{xx} = D_{yy} = D_{zz} = 1000\) Ns/m for the controller damping, and \(B_\theta = 0.1592\) kg*m² for inertia scaling (i.e. no negative joint torque feedback).
Figure 5.11 End-effector positions/references using the Cartesian impedance controller

The maximum value of the absolute error in Cartesian coordinates is approximately 0.000584 m in the x-direction, 0.000413 m in the y-direction, and 0.00116 m in the z-direction. It can be observed that the tracking errors converge to zero in the parts where the manipulator comes to rest. This tracking error level is practically acceptable compared to the errors that can originate from the defects such as play in the transmission mechanism.

Figure 5.12 End-effector tracking errors using the Cartesian impedance controller

The third simulation is related to making point to point movements using the joint space approximation of the Cartesian space impedance control law. The results of this simulation are presented in Figure 5.13 in terms of the joint coordinates, and in Figures 5.14 and 5.15 in terms of the Cartesian coordinates. The straight lines in the left of Figure 5.13 are related to
the link-side positions and the dash-dot lines are the setpoint signals, the error between these two signals are plotted on the right side of the figure.

The controller parameters which are used during this simulation are $K_x = K_y = K_z = 35000 \text{ N/m}$ for the controller stiffness, $D_\theta = 500 \text{ Nms/rad}$ for the controller damping, and $B_\theta = 0.1592 \text{ kg\cdotm}^2$ for inertia scaling (i.e. no negative joint torque feedback). The maximum value of the absolute error is approximately 0.000738 rad in the first and 0.00166 rad in the second joint.
The maximum value of the absolute error in Cartesian coordinates is approximately 0.00058 m in the x-direction, 0.000414 m in the y-direction and 0.00116 m in the z-direction. It can be observed that the tracking errors converge to zero when the manipulator comes to rest. When the results given in Figures 5.12 and 5.15 are compared, it can be concluded that the tracking errors achieved using the joint space approximation of the Cartesian space impedance control are similar to ones achieved using the Cartesian space impedance control. Thus this modification of the Cartesian control law can be used if it is preferred to use the joint variables for control.

5.2.2. Constrained Motion Control

In this section, the simulation results related to constrained motion control are introduced. First the results related to pressing on a planar surface are given, then the results related to sliding on the same surface while pressing are presented. The design of the desired trajectories for constrained motion slightly differs from the ones related to free motion, since the desired forces in the constrained directions also need to be taken into account. A flat planar surface shown in Figure 5.1 is used in the simulations, thus the contact normal is in y direction of the base frame, i.e. \( n = [0 \ 1 \ 0]^T \). The stiffness and damping coefficients that are used in calculation of the reaction force are \( k_E = 10^5 \text{ N/m} \), \( b_E = 25 \text{ Ns/m} \), respectively. The y-coordinate of the environment surface is \( y_E = 0.49497 \text{ m} \).

The first simulation is related to pressing down on the planar surface. The end-effector starts exactly from the top of the surface. During the simulations only two directions, namely x and y, of the Cartesian control law are controlled. The end-effector is commanded to exert 10 N by moving 0.1 m through the surface without moving sideways. The end-effector positions recorded during this simulation are shown in Figure 5.16. The straight lines are related to the recorded end-effector positions calculated using the forward kinematics, and the dash-dot lines are the reference Cartesian motions. The straight lines shown in Figure 5.17 are the recorded force signals and the dash-dot lines are the desired force setpoints.
Figure 5.16 End-effector positions for pressing on a flat object

The controller parameters used in this simulation are $K_{xx} = 30000 \text{ N/m}$, $K_{yy} = 100 \text{ N/m}$ for the controller stiffness, $D_{xx} = 1000 \text{ Ns/m}$, $D_{yy} = 25 \text{ Ns/m}$ for the controller damping, and $B_\theta = 0.0796 \text{ kg*m}^2$ (i.e. $B_\theta = B/2$) for inertia scaling. The controller stiffness in the y-direction is selected much smaller compared to the stiffness in the x-direction, since y-direction is the constrained direction and forces need to be controlled in that direction.

The reference trajectories in this simulation are similar to point-to-point trajectories used in the previous section. The errors in the end-effector positions and forces along the constrained and unconstrained directions are presented in Figure 5.18. It can be observed that both the...
Figure 5.18 End-effector force/position errors for pressing on a flat object

Errors in position and force are very small when the manipulator comes to rest, but are not zero. This comes from the assumption on the steady state, neglecting the joint friction effect,

\[
g(q_{eq}) = K(\theta_{eq} - q_{eq}) - J^T(q_{eq})k_Enn^T[x(q_{eq}) - x_E] \tag{5.5}
\]

\[
J(\theta_{eq})^T K_x(x(\theta_{eq}) - x_{ref}) + K(\theta_{eq} - q_{eq}) = \bar{g}(\theta_{eq}) \tag{5.6}
\]

of the closed-loop dynamics,

\[
M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau - J^T(q)k_Enn^T[x(q) - x_E] \tag{5.7}
\]

\[
B_\theta\ddot{\theta} + J(\theta)^T[K_x\ddot{x}(\theta) + D_x\dot{x}(\theta)] + \tau = \bar{g}(\theta) \tag{5.8}
\]

It has been assumed that for small displacements, \( J(q_{eq}) \approx J(\theta_{eq}) \), \( x(q_{eq}) \approx x(\theta_{eq}) = x_{eq} \) and \( g(q_{eq}) \approx \bar{g}(\theta_{eq}) \). The second simulation is related to sliding on the planar surface while pressing on it. The end-effector starts exactly at the top of the surface. In this simulation the end-effector is commanded to exert 10 N by moving 0.1 m through the surface, and slide 0.1 m on it in the x-direction. The end-effector position and forces recorded during the simulation are displayed in Figure 5.19 and 5.20, respectively. The straight lines in both figures are related to the end-effector positions calculated using the forward kinematics and dash-dot lines are the reference trajectories.
The controller parameters which are used during this simulation are $K_{xx} = 30000 \text{ N/m}$, $K_{yy} = 100 \text{ N/m}$ for the controller stiffness, $D_{xx} = 1000 \text{ Ns/m}$, $D_{yy} = 25 \text{ Ns/m}$ for the controller damping, and $B_0 = 0.0796 \text{ kg*m}^2$ for inertia scaling. The errors in the end-effector positions and forces for constrained and unconstrained directions are presented in Figure 5.21. It can be observed that both the errors in position and force are very small when the manipulator comes to rest, but are not zero. This comes from the assumption on the steady state (5.5) – (5.6) of the closed-loop dynamics. Nevertheless, the achieved errors are at a practically acceptable level.
5.3. **Summary**

In this chapter, first an impedance identification simulation based on a simulated scenario has been introduced. The simulated scenario has consisted of three consecutive phases, free motion, constrained motion and finally free motion. The results obtained with identification algorithm are in agreement with the defined simulation environment properties. Then free motion control simulations have been performed by using the joint space impedance control, Cartesian space impedance control and joint space approximation of Cartesian controller. It has been shown that the tracking error performance of Cartesian impedance control and its joint space approximation are similar for free motion case. Simulations related to constrained motion control have also been conducted by using the Cartesian impedance controller. Two different simulations have been performed where the first one is related on pressing on a planar object and the second one is sliding on the object while applying a force on it.
6. Experimental Results

In this chapter, first a brief introduction of the experimental setup is given, and then the results of the identification experiments based on the method explained in chapter 3 are presented. Finally, the results obtained using the impedance controller described in chapter 4 are shown. The identification experiments are performed by means of the programming by demonstration (PbD) framework. During the experiments, the user holds the robot arm at the wrist in order to guide the arm and apply forces to the interacted environment. The demonstrations are conducted when the robot is operated in the zero-gravity mode. In this mode only the gravity torques are applied to the motors, and the robot arm floats freely in the air.

6.1. Properties of the experimental setup

The experiments presented in this thesis are conducted on a 2 degrees-of-freedom robot arm which is shown in Figure 6.1. The directions of the base frame (i.e. X-Y-Z) together with the positive directions of the joint angles (i.e. $q_1$-$q_2$) are also indicated in this figure. The kinematics of this structure can be found in the Appendix.C. The properties related to the mechanical transmission, electronic equipment, and also the sampling frequency used during these experiments are given in Table 6.1. A differential drive mechanism is utilized in order to achieve the two degrees-of-freedom of the robot arm. Two DC motors are used to actuate these two degrees-of-freedom. The actuation power of the DC motors is transmitted to the differential drive by using a cable transmission.

The arm is equipped with incremental encoders connected to the motors, torque sensors measuring the joint torques, and a 6 d.o.f. force/torque (F/T) sensor which is capable of measuring the forces and moments applied at the tip of the end-effector. A stiff pin-like object is also mounted at the force sensor in order to achieve a point contact between the end-
effector and the environment. Information about the calibration of the joint torque sensors and the 6 d.o.f. force/moment sensor can be found in the Appendix D.

Table 6.1 Parameters of the experimental setup

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling time, $\Delta T$ [ms]</td>
<td>0.001</td>
</tr>
<tr>
<td>Gearbox ratio, $i_g$</td>
<td>1:66</td>
</tr>
<tr>
<td>Cable transmission ratio, $i_{ct}$</td>
<td>101.5/31.5</td>
</tr>
<tr>
<td>Total transmission ratio, $i_{total}$</td>
<td>212.67</td>
</tr>
<tr>
<td>Motor encoder resolution (cnt/rev)</td>
<td>2000</td>
</tr>
<tr>
<td>Motor inertia, $B_m$ [kg*m$^2$]</td>
<td>$34.5*10^{-7}$</td>
</tr>
<tr>
<td>Gear inertia, $B_g$ [kg*m$^2$]</td>
<td>$0.7*10^{-7}$</td>
</tr>
<tr>
<td>Total inertia reflected to link side</td>
<td>$0.1592$</td>
</tr>
<tr>
<td>DC Motor torque constant, $K_c$ [mNm/A]</td>
<td>53.8</td>
</tr>
<tr>
<td>Amplifier constant, $K_{vc}$ [A/V]</td>
<td>0.389</td>
</tr>
</tbody>
</table>

The parameters of the DC motors are obtained from the catalog [33] and are given in the Appendix E. The voltages applied to the power amplifiers are calculated in the software as follows:

$$V_{D/A} = \frac{r_m}{i_{total}} \cdot \frac{1}{10^3 K_c} \cdot \frac{1}{K_{vc}}$$  \hspace{1cm} (6.1)

Here, parameters given in Table 6.1 are used. The angular velocities are determined from the motor encoder measurements by means of numerical differentiation [34]:

$$\bar{v}(j) = \frac{1}{n} \sum_{i=0}^{n-1} v(j-i) = \frac{1}{n \cdot \Delta T} (q(j) - q(j-n))$$  \hspace{1cm} (6.2)

where $q(j)$ represents the encoder measurement at the current time sample, $q(j-n)$ is the $n$-sample previous encoder reading, and $\Delta T$ is the sampling time. Practically, (6.2) implies that the last $n$ velocity estimates are simply averaged. In all of the experiments conducted, $n = 10$ is observed to be sufficient to reduce the effect of the measurement noise.

The real-time control interface that is used to control the robot arm is through a dSPACE DS1103 block. The controllers, reference setpoints and any other controller related features are designed in Matlab/Simulink and then transmitted to this interface in order to be executed in real-time.

6.2. Impedance Identification

In this section the experimental identification of the contact impedance (also called contact dynamics) is presented. Three different experiments are conducted, where two are related to interaction with a stiff and flat (nearly rigid) object (a liftboy), and the third one is related to interaction with a soft object (a sponge). The results of the third experiment which is related to interaction with the soft object (i.e. sponge) is given in Appendix F.2. The end-effector
positions and forces recorded during these experiments are low-pass filtered using a 4th order Butterworth filter with a cut-off frequency of 50 Hz. The arm has only two rotational d.o.f.’s, so it is not possible to excite the environment in all three directions independently. Therefore, the y-direction is selected to be the inherent motion direction, and the impedance perception is not performed in that direction. The first experiment is related to pressing down on the stiff object. The demonstrated task consists of three phases:

1) **Unconstrained motion**: Guiding the arm from its initial posture through free space until its tip makes a contact with a horizontally placed stiff object (which is parallel to the X-Y plane indicated in Figure 6.1).
2) **Constrained motion**: Pressing down with the end-effector on the surface to apply some force, and finally,
3) **Unconstrained motion**: Bringing the arm back to its initial posture.

The end-effector forces recorded during the demonstrations can be observed in Figure 6.2. It can be noticed that when the tip is pressed down, the force in the z-direction displays a significant change compared to the forces in other directions.

The end-effector positions recorded during the demonstrations can be observed in Figure 6.3. These are calculated by using forward kinematics with the measured encoder readings.
The parameters of the virtual soft finger (V.S.F.) which are used during the experiments are selected as $K_v = 700 \, N/m$ and $B_v = 10 \, Ns/m$. The parameters of the forgetting factor are selected as $X_H = 0.02 \, m$ and $T_H = 1 \, sec$. The other parameters that are used during the experiments are $A = 0.5$ for the uncertainty estimation parameter and $U = 0.1$ for the discontinuity detection parameter. The norm of the end-effector velocity is given in Figure 6.4. Since only the x and z components of both force and position data have been used, the norm of the velocity is calculated from differentiating the x and z position data after the virtual soft finger algorithm is applied. According to the selection of the forgetting factor parameters, the bound on the low-speed motion becomes $X_H / T_H = 0.02 \, m/s$.

The identification results achieved using the impedance perception algorithm are given in
Figures 6.5, 6.6 and 6.7. The straight lines in these figures represent the estimated components of the impedance determined using the recursive algorithm, the horizontal dash-dot lines are the estimates achieved using the batch least squares algorithm, the gray bands represent the uncertainties of the estimates, and the vertical dotted lines indicate the instants when the discontinuities are detected (i.e. gain and loss of contact in this case). The batch processing is applied when the end-effector is in constrained motion phase. The bias terms that are estimated by the impedance perception algorithm are given in Figure 6.5.

The estimated discontinuity instants are close to the actual moments when the discontinuities occur. This can be verified based on the end-effector force plots shown in Figure 6.2, since at 9.7 seconds the end-effector forces become different than zero and start increasing, while at 20.9 seconds the end-effector forces become zero again.

Figure 6.5 Estimated bias terms by recursive/batch algorithms, uncertainties, and discontinuities

Figure 6.6 Estimated stiffness terms using recursive/batch algorithms and the uncertainties
Here the discontinuities are detected using the method introduced in section 3.3.4, which is based on the difference between past and present model outputs. The stiffness terms determined using the impedance estimation algorithm are given in Figure 6.6. It can be observed that when the end-effector is in contact with the liftboy, the stiffness $K_{zz}$ converges to $K_v$ with a small uncertainty. This is because the surface normal is in the z-direction and the object is very stiff. The stiffness $K_{xx}$ in the tangential direction, when the end-effector is in contact with the liftboy, converges also close to $K_v$ with a relatively small uncertainty. This is the most likely because of the static friction between the end-effector and the surface during the pressing phase. This is also confirmed from the force plots $F_x$ in Figure 6.2, where one can observe that the force is varying whereas the position in $x$ direction is constant. It can also be observed that the off diagonal terms converge close to the correct zero values. There are also some transient effects because of the resetting of the algorithm at transitions from the unconstrained to constrained motions and vice versa. These transient effects vanish approximately in 1 second. After the transients, both the bias and the stiffness estimates in both of the unconstrained motion phases are close to zero with a low uncertainty. This can of course be expected, since there is no reaction force coming from the environment in these segments. The liftboy surface’s z coordinate at the contact point based on the end-effector position measurements is approximately -0.4566 m, which can be observed in Figure 6.3. The consistency of the bias and stiffness terms in z-direction can also be checked by calculating the absolute value of this coordinate by using the values obtained from batch processing in constrained motion phase as follows

$$\frac{c_z}{K_{zz}} = \frac{315.3}{691.9} = 0.4557 \text{[m]}$$

which is very close to the measured value. The damping terms that are obtained by the impedance estimation algorithm are given in Figure 6.7. It can be observed that the uncertainties of the damping terms are relatively higher, which is the most probably due to the fact that the input signals are not sufficiently persistently exciting.

![Figure 6.7 Estimated damping terms by recursive/batch algorithms and corresponding uncertainties](image)

Here the input signals are the inputs of the model
where $k$ denotes the current time instant and $\psi_k = \begin{bmatrix} 1 & p_k^T & p_{k-1}^T & p_{k-2}^T \end{bmatrix}^T$ contains the end-effector positions shown in Figure 6.3. The related mass matrix terms that are obtained by the impedance estimation algorithm are given in the Appendix F.2. The estimator’s performance can also be checked by comparing the outputs (i.e. forces) of the model given by (6.3) with the measured (raw) outputs. The estimated output is determined using the following relationship:

$$\hat{\phi}_k = \hat{\Theta}_k^T \psi_k$$

The measured outputs, the model output (i.e. the estimate) and their mutual difference are given in Figure 6.8. It can be observed that the model outputs are matching the sensed signal sufficiently well except when discontinuities occur. The effect of these discontinuities in the estimated impedance parameters and the output are present only for a short period of time.

The second experiment is related to sliding over a stiff object. The demonstrated task consists again of three phases:

1) *Unconstrained motion*: Guiding the arm from its initial posture through free space until its tip makes a contact with a horizontally placed stiff object (which was parallel to the X-Y plane indicated in Figure 6.1).
2) *Constrained motion*: Making a wiping motion over the surface of the object while at the same time applying a force on it, and finally,
3) *Unconstrained motion*: Bringing the arm back to its initial posture.

The end-effector positions and end-effector forces recorded during the demonstrations are shown in Figure 6.9. It can be observed that the force in the $z$-direction exhibits a significant change compared to the forces in other directions while the tip is in contact.
The effect of the contact friction while the tip is contact with the object can be observed from Figure 6.11. Here, the vertical dotted lines represent the discontinuity instants (i.e., establishing and leaving contact). It can be observed that the pattern of the force in the $x$-direction, while the tip is in contact, is more or less concurrent with the motion (i.e., velocity) of the tip. The parameters of the virtual soft finger (V.S.F.) that are used during the experiments are $K_v = 700 \text{ N/m}$ and $B_v = 10 \text{ Ns/m}$. The parameters of the forgetting factor are $X_H = 0.02 \text{ m}$ and $T_H = 1 \text{ sec}$. The uncertainty estimation parameter that is used during the experiment is $A = 0.5$. The norm of the end-effector velocity is given in Appendix F.2.
The discontinuities (i.e. gain and loss of contact) are detected using the thresholding on the norm of the force, instead of using the second method mentioned in section 3.3.4. This is because the second method is oversensitive to unmodeled friction effects and detects more discontinuous instants while the tip is in contact with the object. The norm of the end-effector forces is given in Figure 6.12. The threshold on the force is selected as 0.53 [N]. The estimated discontinuity instants are close to the actual moments when the discontinuities occur. This can also be seen from the end-effector force plots in Figure 6.9, since after 6 seconds the end-effector forces become different than zero and start increasing, while after 16.9 seconds the end-effector forces become zero again.

The stiffness terms that are obtained using the impedance estimation algorithm are given in Figure 6.13. The straight lines in these figures represent estimated stiffnesses determined.
using the recursive algorithm, the horizontal dash-dot lines are the estimates determined using the batch least squares algorithm, the gray bands represent the uncertainties of the estimated impedance components. The batch processing is applied when the end-effector is in constrained motion phase.

![Figure 6.13 Stiffness terms estimated using the recursive/batch algorithms and uncertainties](image)

It can be observed that when the end-effector is in contact with the lifboy the stiffness value $K_{zz}$ is close to $K_v$ with a small uncertainty. This is because the surface normal is in the $z$-direction and the object is very stiff. The stiffness in the tangential direction component $K_{xx}$, when the end-effector is in contact with the lifboy, is also very close to zero with a relatively small uncertainty. It can also be observed that the off diagonal term $K_{xz}$ converges close to the correct zero value, while the other $K_{zz}$ fluctuates with a relatively large uncertainty, most probably due to kinetic friction acting on the tip while it slides over the object’s surface. Both the bias and the stiffness values in both of the unconstrained motion phases are close to zero with a low uncertainty, excluding the short transients. The related bias, damping and mass matrix terms that are obtained using the impedance estimation algorithm are given in the Appendix F.2. The estimator’s performance can also be checked by comparing the outputs (i.e. forces) of the model given by (6.3) with the measured (raw) outputs. The measured outputs, the estimated output and their difference are given in Figure 6.14 for the 1st element and in Figure 6.15 for the 2nd element, respectively. It can be observed that the estimator’s outputs for the 2nd element of $\hat{\phi}_z$ (i.e. force in z direction) matches the sensed signal quite well, whereas for the 1st element this error is relatively higher. This is mainly because the contact model does not contain the effect of contact friction.
6.3. Results of Control Experiments

6.3.1. Unconstrained Motion Control

In this section, the experimental results related to free motion control are introduced. First, the results related to joint space position control given by (4.9) – (4.10) are presented, then the results related to Cartesian space position control (4.23) – (4.24) and its joint space approximation (4.23) – (4.30) are introduced. As mentioned before, the arm which is used during the experiments has only 2 d.o.f. obtained by a differential drive. It is also mentioned that the workspace of the robot is a nearly spherical 2D surface in the Cartesian space and
therefore the end-effector can be placed only at the points on this surface. Consequently, same strategy for the design of the trajectories and control in the Cartesian space case is used as in simulation case-studies presented in Chapter 5. Since a sufficiently accurate value of the joint elasticity was not available, the gravity compensation term \( g(q) = g(\vec{q}(\theta)) = \vec{g}(\theta) \), mentioned in section 4.1.1, is not used in the experiments. Instead, we apply \( g(\theta) \), which does not take the elasticity into account. When such a gravity compensation term is applied, the gravity torques in the steady-state are not compensated perfectly, and therefore some steady-state errors remain. The gravity compensation term \( g(\theta) \), is also model based, therefore its accuracy depends on the parameters that are taken into account. Two experiments are conducted for joint space impedance control: in the first one, no negative joint torque feedback is used while in the second it is used. The trajectory (position, velocity and acceleration) which is used during these experiments is shown in Figure 6.16. It consists of two trapezoidal velocity profile trajectories, which brings the joint to its initial position at the end of the trajectory. The trapezoidal trajectory contains a constant velocity phase following the constant acceleration phase and ending with a constant deceleration phase. This trajectory is introduced repetitively to investigate tracking performance of the controller. Here, the inertia scaling coefficient is taken equal to the total rotor inertia reflected to the link side, as given in Table 6.1.

![Figure 6.16](image)

Figure 6.16 The setpoint trajectory in tracking experiments

The parameters of the controller that are used in the experiment without joint torque feedback are given in Table 6.2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint stiffness, ( K_\theta ) [Nm/rad]</td>
<td>9000</td>
</tr>
<tr>
<td>Joint damping, ( D_\theta ) [Nms/rad]</td>
<td>50</td>
</tr>
<tr>
<td>Inertia scaling, ( B_\theta ) [kg*m²]</td>
<td>0.1592</td>
</tr>
</tbody>
</table>

The joint stiffness and damping values are selected as high as possible, in order to obtain low tracking error and maintain stability in the presence of the measurement noise, unmodeled
high frequency dynamics, etc. The results of this experiment are depicted in Figure 6.17. The straight lines on the left hand side are related to the recorded encoder signals, and the dashed lines are the reference trajectory signals, and the error between these two signals are plotted on the right hand side of the figure.

The maximum value of the absolute error in this experiment is approximately 0.00163 rad for the first and 0.00166 rad for the second joint. This servo error level introduces typical displacement errors of approximately 1 mm at the end-effector. This is an acceptable error level compared to the errors originating from the play in the transmission mechanism. It can be observed from Figure 6.17 that high tracking errors occur when there is a direction change in the setpoint, which is most likely caused by the friction in the actuators and other parts of the transmission mechanism. The parameters of the controller that are used in the experiment with joint torque feedback are given in Table 6.3. Here, the inertia scaling coefficient is selected about a half of the total rotor inertia reflected to link side given in Table 6.1, thus it introduces negative joint torque feedback.

### Table 6.3 Joint space controller parameters with torque feedback

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint stiffness, $K_\theta$ [Nm/rad]</td>
<td>9000</td>
</tr>
<tr>
<td>Joint damping, $D_\theta$ [Nms/rad]</td>
<td>50</td>
</tr>
<tr>
<td>Inertia scaling, $B_\theta$ [kg*m$^2$]</td>
<td>0.07</td>
</tr>
</tbody>
</table>

The results of this experiment are depicted in Figure 6.18. The straight lines on the left hand side are related to the recorded encoder signals, and the dashed lines are the reference trajectory signals, and the error between these two signals are plotted on the right hand side of the figure.
The maximum value of the absolute error in this experiment is approximately 0.00074 rad for the first and 0.00072 rad for the second joint. The effect of the negative joint torque feedback on reducing the tracking error can be observed when the errors in Figures 6.17 and 6.18 are compared. The joint torque feedback coefficient has been decreased approximately 2.3 times and the maximum value of the absolute error has reduced about 2.2 times for the first and 2.3 times for the second joint. Consequently, it can be concluded that reducing the inertia scaling coefficient introduces negative joint torque feedback, which in turn reduces the effect of the aforementioned friction effects.

The second experiment is related to making point to point movements by using the Cartesian space impedance control law. The point to point trajectories for the Cartesian coordinates can be generated in many ways, such as by means of cubic or, quintic polynomials, splines, etc [37]. The point to point trajectories which are used during these experiments have been generated the same way that are done in the simulations explained in section 5.2.1. The end-effector is commanded to make consecutive point-to-point movements in the horizontal x-y plane in this experiment. These trajectories are selected in the x-z directions and the trajectory in the y-direction is calculated using the robot forward kinematics. The results of this experiment are shown in Figure 6.19 and 6.20. The straight lines shown in Figure 6.19 are related to the actual end-effector positions (calculated using the forward kinematics based on encoder measurements), the dashed lines are the reference trajectories, and the error between these are plotted in Figure 6.20. The controller parameters which are used during this experiment are $K_{xx} = K_{yy} = K_{zz} = 8000 \text{ N/m}$ for the controller stiffness, $D_{xx} = D_{yy} = D_{zz} = 40 \text{ Ns/m}$ for the controller damping, and $B_\theta = 0.08 \text{ kg} \cdot \text{m}^2$ for inertia scaling.
The maximum value of the absolute error in Cartesian coordinates is approximately 0.0017 m in the x-direction, 0.000873 m in the y-direction, and 0.000849 m in the z-direction. It can be observed from this figure that the errors do not converge to zero when the manipulator comes to rest. The reasons for this are the remaining static friction in the transmission mechanism and imperfect gravity compensation.

The last experimental result for the unconstrained motion control case is related to following a trajectory demonstrated within the PbD framework. During this experiment the arm is moved within the workspace and the encoder signals are saved in the computer. These recorded data are used later as reference trajectories for the controller. In this experiment, the joint space approximation of the Cartesian control law is used. The encoder signals are recorded during
this experiment are low-pass filtered offline and downscaled by a factor of 10 before replay. The downsampling of the saved data is performed because of the memory limitations of the control interface block. A second order Butterworth filter (implemented with the Matlab command “butter”) with a cut-off frequency of 20 Hz is used to filter the recorded signals. Since this processing is performed offline, the Matlab routine “filtfilt” is also used to avoid phase delays. The angular joint velocities accelerations, that are used in the feedforward components of the control law, are obtained by numerical differentiation. These are presented in Figure 6.21.

It can be observed that the velocities are not as smooth as the hand-designed trajectories in the former experiments, whereas the accelerations are far less smoother. This limitation can be overcome by using the trajectory learning techniques that are mentioned in Chapter 2. The controller parameters that are used for replay of the recorded trajectory are $K_{xx} = K_{yy} = K_{zz} = 9000 \text{ N/m}$ for the controller stiffness, $D_\theta = 50 \text{ Nms/rad}$ for the controller damping, and $B_\theta = 0.1592 \text{ kg*m}^2$ (i.e. no negative joint torque feedback) for inertia scaling. The joint coordinates and tracking errors obtained in this experiment are presented in Figure 6.22, and Cartesian coordinates and tracking errors are shown in Figure 6.23 and 6.24 respectively. The straight lines on Figure 6.23 and on the left hand side of Figure 6.22 are related to the recorded encoder signals, the dashed lines are the reference trajectory signals. The error between these two signals are plotted on Figure 6.24 and on the right hand side of the Figure 6.22.
The maximum value of the absolute error in this experiment is approximately 0.0066 rad for the first and 0.00358 rad for the second joint. The maximum value of the absolute error in Cartesian coordinates for this experiment is approximately 0.00237 m in the x-direction, 0.00205 m in the y-direction, and 0.00295 m in the z-direction. This tracking error is slightly larger as compared to the former experiments. This can be seen, for example when the errors in figures 6.17 and 6.22 are compared. The reasons for this are non-smoothness of velocity and acceleration feedforward signals, friction in the transmission mechanism, and imperfect gravity compensation. Nevertheless, the performance that is achieved in this experiment is sufficient for replaying a human demonstration of a domestic task to the robot.
6.3.2. Constrained Motion Control

In this section, the experimental results related to constrained motion control and contact transition control are presented. First the results related to sliding on a stiff surface while pressing are introduced, and then the execution of contact transitions by replaying a demonstrated task are presented. The results of an additional experiment related to pressing on a stiff surface are given in Appendix F.2. The design of the desired trajectories for constrained motion slightly differs from the ones related to free motion, since the desired forces in the constrained directions also need to be taken into account. The steady-state values of the end-effector positions and forces have been derived for constrained motion control in section 4.2. These can also be used for the design of the corresponding reference trajectories for the constrained and unconstrained directions. When the friction terms are neglected, the end-effector position at steady state, given by (4.49) can be rewritten as,

$$x_{eq} = nn^T \left[ \frac{k_f}{k_E + k_f} \left( x_{ref} - \frac{1}{k_f} \delta_c \right) + \frac{k_E}{k_E + k_f} x_E \right] + \left( I - nn^T \right) \left[ x_{ref} - \frac{1}{k_p} \delta_c \right]$$

$$x_{eq} \approx \frac{1}{k_E + k_f} nn^T \left( k_f x_{ref} + k_E x_E \right) + \left( I - nn^T \right) x_{ref} \tag{6.5}$$

Since it is important to track the desired position setpoints along the unconstrained directions, the position reference for the unconstrained directions can be selected as,

$$x_{ref} = x_{eq} \tag{6.6}$$

On the other hand, when the friction terms are neglected, the end-effector force at steady-state, given by (4.50), can be rewritten as,
\[ f_{\text{est}, \text{eq}} = \frac{k_E k_f}{k_E + k_f} nn^T \left[ x_{\text{ref}} - \frac{1}{k_f} \delta_{\infty} - x_E \right] \]

\[ f_{\text{est}, \text{eq}} \approx \frac{k_E k_f}{k_E + k_f} nn^T \left( x_{\text{ref}} - x_E \right) \]  

(6.7)

For very stiff environments \( k_E \gg k_f \), the equivalent stiffness can roughly be approximated by \( \frac{k_E k_f}{k_E + k_f} \approx k_f \), and (6.7) simplifies to,

\[ f_{\text{est}, \text{eq}} \approx k_f nn^T \left( x_{\text{ref}} - x_E \right) \]  

(6.8)

Since it is important to track the desired force setpoints in the constrained directions, the position reference that achieves the desired force can be calculated from (6.8) as,

\[ x_{\text{ref}} = x_E + \frac{f_{\text{est}, \text{eq}}}{k_f} \]  

(6.9)

The first result is related to sliding on a stiff surface while pressing on it. The end-effector is again brought on top of the stiff surface in zero-gravity mode, and then it is switched to the Cartesian control mode to execute the experiment. Two experiments are conducted for sliding on a stiff surface task. In the second experiment, the stiffness of the unconstrained direction (i.e. x-direction) is selected higher as compared to the first. This is done in order to check whether the tracking error in the unconstrained directions would improve. In both sliding experiments the end-effector is commanded to exert 10 N by moving 0.1 m through the object, slide 0.05 m on it, and return back. The end-effector forces and positions recorded during the experiment are displayed in Figure 6.25 and 6.26 respectively.

![End effector forces](image-url)
The controller parameters that are used during the first sliding experiment are $K_{xx} = 2000 \, N/m$, $K_{zz} = 100 \, N/m$ for the controller stiffness, $D_{xx} = 10 \, Ns/m$, $D_{zz} = 5 \, Ns/m$ for the controller damping, and $B_{\theta} = 0.025 \, kg\cdot m^2$ for inertia scaling. The errors in the end-effector positions and forces for along the constrained and unconstrained directions are presented in Figure 6.27. The maximum value of the absolute error in Cartesian coordinates is about 0.0031 m in the x-direction. The maximum value of the absolute error for force tracking is about 3.89 N in the z-direction. The errors during the steady-state part of the trajectory between 10 and 18 seconds is 0.00264 m in the x-direction and 1.75 N in the z-direction.

The end-effector positions and forces recorded during the second sliding experiment are given in the Appendix F.2, in figures F.19 and F.20 respectively. The controller parameters that are
used during this experiment are $K_{xx} = 3000 \text{ N/m}$, $K_{zz} = 100 \text{ N/m}$ for the controller stiffness, $D_{xx} = 20 \text{ Ns/m}$, $D_{zz} = 5 \text{ Ns/m}$ for the controller damping, and $B_\theta = 0.025 \text{ kg*m}^2$ for inertia scaling. The errors in the end-effector positions and forces for the constrained and unconstrained directions are presented in Figure 6.28. The maximum value of the absolute error in Cartesian coordinates is about 0.00219 m in the x-direction. The maximum value of the absolute error for force tracking is about 3.38 N in the z-direction. The errors during the steady-state part of the trajectory between 10 and 18 seconds is approximately 0.00176 m in the x-direction and 1.76 N in the z-direction.

When Figures 6.27 and 6.28 are compared, it can be observed that increasing the Cartesian stiffness in x-direction results in reduction of the tracking error as expected. The force tracking errors between the two experiments are nearly the same, since the stiffness in the z-direction is not changed. The reasons for this are the static friction in the transmission mechanism and imperfect gravity compensation. The friction between the end-effector and the surface might also effect the tracking error in the unconstrained direction.

The final experiment is related to executing the constrained motion control including contact transitions within the PbD framework. The phases of the demonstrated task are the same as in the identification experiments related to sliding on the stiff object. The position trajectory in x-direction in both free and constrained motion phases is selected as setpoint to the controller. The position trajectory in z-direction is used as the setpoint to the controller in free motion phases, whereas in constrained motion phase, a combination of position and force trajectories according to (6.9) is used as the setpoint. During the execution, the object surface is lowered down for approximately 0.02 m, in order to check the effect of position uncertainties on the transition algorithm. The demonstrated trajectories that are used during this experiment are low-pass filtered offline and downsampled by a factor of 10 before replay. A second order Butterworth filter with a cut-off frequency of 20 Hz is used to filter the recorded position signals, and a first order Butterworth filter with a cut-off frequency of 5 Hz is used to filter the recorded force signals. The Matlab routine “filtfilt” is used again to avoid phase delays during filtering. The end-effector forces and positions recorded during the experiment are shown in Figure 6.29 and 6.30 respectively. The straight lines represent the forces and positions.
recorded during replay, the dashed lines are the reference trajectories required to execute the task, and the dotted lines are the trajectories recorded from the demonstrations. The controller parameters for the unconstrained motion phases of the replay are $K_{xx} = 4000 \text{ N/m}$, $K_{zz} = 4000 \text{ N/m}$ for the controller stiffness, $D_{xx} = 60 \text{ Ns/m}$, $D_{zz} = 60 \text{ Ns/m}$ for the controller damping, and $B_\theta = 0.1274 \text{ kg*m}^2$ for inertia scaling. The controller parameters for the constrained motion phase that are used during the replay are $K_{xx} = 2000 \text{ N/m}$ and $K_{zz} = 150 \text{ N/m}$ for the controller stiffness, $D_{xx} = 30 \text{ Ns/m}$ and $D_{zz} = 20 \text{ Ns/m}$ for the controller damping, and $B_\theta = 0.091 \text{ kg*m}^2$ for inertia scaling. The strategy explained in section 4.3 is used during the transition from free to constrained motion and vice versa, and the gains are interpolated smoothly but quickly instead of a stepwise change. The upper bound of the hysteresis on the force signals is $f_{\text{hres,high}} = 2.5 \text{ N}$, whereas the lower bound is $f_{\text{hres,low}} = 0.5 \text{ N}$.

![Figure 6.29](image1.png)  
End-effector forces for contact transition control on a stiff object

![Figure 6.30](image2.png)  
End-effector positions for contact transition control on a stiff object
It can be observed from Figure 6.31 that when the end-effector makes contact with the surface at 12.4 second, the z component of the force shows a large peak of approximately 31 N. Duration of this peak is quite short (about 0.02 seconds). Furthermore, the end-effector does not bounce continuously on the surface because of any transition instability. This shows that the control method is capable of executing transitions between different phases. The errors in the end-effector positions and forces for the constrained and unconstrained directions are presented in Figure 6.31. The position tracking performance is reasonable, unlikely the force tracking. There are mainly two reasons for this. The first one is related to the fact that the object has been moved down. As mentioned before, the desired force using impedance control can be achieved by properly selecting the stiffness and desired position setpoint inside the environment in the constrained direction. Therefore when the object is moved down the desired deflection given inside the environment to achieve the amount of force similar to that of the demonstrations is reduced. The second reason is related to the setpoint which is applied in the constrained motion phase. In order to obtain a force trajectory in the constrained direction, similar to that of the human demonstrations shown in Figure 6.9, the position trajectory given inside the environment is calculated by using (6.9). However it can be observed from the force plot in the z-direction in Figure 6.9, that the recorded force signals are a lot more noisy compared to position signals. Therefore in order to better replay the demonstrated forces, a more advanced signal processing method or even a trajectory learning method should be implemented to create smoother setpoints in the constrained motion phase for the impedance controller.

![Figure 6.31](image)

**Figure 6.31** End-effector force/position errors for contact transition control on a stiff object

### 6.4. Summary and Comparison

In this chapter two impedance identification experiments have been introduced. The first one is related to pressing on a stiff object whereas the second one is related to sliding on it. The effect of static and kinetic contact friction on the identified stiffnesses has been shown. Two different discontinuity detection approaches to detect transitions between free and constrained motions have been used. The one which is based on detecting the changes between past and
present models is found to be too sensitive to unmodeled contact friction. Free and constrained motion control experiments have been introduced. The impedance controllers which are explained in chapter 4 have been used for this purpose. The replay of two simple tasks has been performed. The first one is related to replaying a demonstrated free motion trajectory and the second one is related to replaying a complete demonstrated task which contains contact transitions and constrained motion. The effect of joint torque feedback on reducing the motor side friction together with the position tracking error has been shown. For the sliding task, increasing the controller stiffness in the unconstrained directions has been helpful to reduce the tracking error in those directions.

Some comparisons can also be made with the simulation results obtained in the previous chapter. When the identification results are compared, it can be observed that the algorithm performs relatively better for the sliding task in the simulations. A reason for this is the fact that contact friction has been present during the experiments, whereas it has not been modeled in simulations. The controller gains which can be selected for free motion control and unconstrained directions of the constrained motion control simulations are quite high compared to the stiffnesses in experiments. In reality such high (about 30000 N/m) stiffness values will not be feasible because of measurement noise (e.g. quantization error in encoders), unmodeled dynamics (e.g. motor friction, play in transmission, etc.) and motor torque saturation. The tracking errors achieved in the simulations is slightly better compared to the experiments. The steady state performance achieved during the simulations is a lot better compared to the experiments. This is mostly because there wasn’t any motor side friction in the simulations and it has been possible to apply the modified gravity compensation law since the joint stiffness is known in simulations. The force control performance is better in the simulations compared to the experiments.
7. Conclusions and Recommendations

7.1. Conclusions

In this master’s thesis, various existing approaches to programming complex robotic manipulation tasks that are particularly related with household applications have been investigated. Two types of methods can be discerned based on this investigation: pre-programming and programming by human demonstration (PbD). The latter is more beneficial for end-users in domestic environments, since it allows robot programming in an easy fashion. This is important since the most home users would lack the knowledge of how to program a robot to perform an intended task. The concept of atomic skills which can be used for formulating a complete domestic task, such as cutting a carrot, has been introduced. It has been realized from that most of these skills require interaction with the environment (i.e. constrained motion tasks) and transitions between free and constrained motions. Therefore, the notion of natural/artificial constraints related to a given task has been introduced for better execution of tasks involving extensive contact with the environment.

A way to execute compliant motion tasks, such as opening a door, has been introduced by using natural/artificial constraints. The concept of contact impedance has been used in order to describe the interaction between the manipulator and the environment. It can be applied both for identifying the natural constraints of the environment and for tuning of the controller in constrained motion phase. A linear contact model, comprised of stiffness, viscosity and inertia terms, has been estimated using a weighted recursive least squares algorithm. The transitions between free and constrained motion phases have been identified by using two methods, thresholding on contact force sensor signals and detecting abrupt changes in the prediction error when a discontinuous change in constraint condition occurs. The identification algorithm has been tested both in simulations and experiments. Three different identification experiments and simulations contained three consecutive phases: unconstrained, constrained and again unconstrained motion. It has been found that the linear model of interaction is not sufficient to describe the behavior of the soft object and the effect of contact friction. It has been realized that the second discontinuity detection method is sensitive to unmodeled contact dynamics, such as contact friction and non-linear stiffness and damping.

Another important issue is ability to control the interaction between the robot and the environment. Impedance control has been selected for this purpose, since it can be used both in free and constrained motion phases. A passivity based approach has been used, since it offers some robustness against unmodeled manipulator and actuator dynamics. Two different types of impedance controllers have been implemented both in simulations and experiments. The difference between these controllers is related to the coordinates in which they are calculated. The first controller is implemented in joint coordinates, whereas the second one is in Cartesian coordinates. The effects of joint flexibility and friction on the control law have been discussed. The advantage of using negative joint torque feedback in combination with impedance control to improve tracking error performance has been shown by experiments. A possible way of tuning the impedance controller and a manner of selecting the corresponding reference trajectories when conducting constrained motion tasks are also introduced. A method that can be used to handle transitions from free to constrained motion and vice versa,
in the presence of stiff environments, has been presented. Replay of very simple demonstrated tasks has also been performed.

7.2. Recommendations

The approaches presented in this thesis can be improved in several ways. First of all, recommendations related to the identification algorithm are given. Then, suggestions about the controller are presented. Finally, some advices about programming by demonstration methods are offered.

Some improvements are possible in the contact impedance identification algorithm. First of all, instead of using the bilinear transformation for discretization in this estimation algorithm, Kalman filters or different observers can be used to estimate the velocity and acceleration, for the sake of better estimation of the damping and the mass matrices. The interacted environments will always contain contact friction; therefore, its effect should be included in the identification algorithm, in order to improve the estimation of the contact stiffness and damping. It is preferable to incorporate a simple contact friction model (e.g. Coulomb friction), since the purpose is to improve the quality of estimated contact stiffness and damping and not making a perfect contact friction model. The discontinuity detection scheme that uses thresholding on force signals can only detect whether there is contact or not. However, this might not be sufficient for contact tasks where multiple contact points are present or when there are transitions between low and very high environment stiffnesses. The idea behind the second discontinuity detection method is more general; however it is not robust to non-modeled effects such as contact friction or nonlinear impedance characteristics. Therefore, further investigation is necessary for designing a both general and robust discontinuity detection method.

Since least squares estimation has been used in the algorithm, it will always be vulnerable to non-persistently exciting signals. Possible ways to cope with non-persistently exciting signals can be using damped least squares, instead of using the simple weighted least squares criterion [40]. This modification would aim at improving the condition number of the regressor matrix which appears in the solution of the least squares algorithm. This is the case since non-persistently exciting signals badly affect the condition number. The damping coefficient can be selected constant or be adapted (which is often done in Levenberg-Marquardt type algorithms). Robust version of recursive least squares (with covariance resetting, [38]) can also be used to deal with non-persistently exciting signals. Other robust adaptive filtering methods can also be investigated for further improvement. A different type of forgetting factor can further be investigated to cope with the nonlinearities of the environment, instead of the one used in the existing algorithm. Possible ways to reduce the high peaks occurring in the estimates after the transitions happen should also be investigated.

In the derivation of the manipulator dynamics, a possible way to incorporate joint damping in parallel to joint flexibility has been presented. This can be used to design controllers that include the derivative of measured joint torque signals as suggested in [26, 27]. The position tracking performance of the implemented impedance controller is at acceptable level; however, its end-effector force tracking performance can still be improved for contact tasks (e.g. assembly) that require better tracking of the end-effector force setpoints. The force sensor at the end-effector can be used for this purpose; however the control would be non-
collocated in that case. The torque sensors at the joints can also be used for this purpose, however the effect of the rigid robot dynamics (i.e. inertia, Coriolis, gravity terms) should be removed from the measured torques in that case. An exact link-side Cartesian stiffness design that takes the effect of joint flexibility into account should also be implemented to improve the response in constrained motion phase. Further investigation is necessary in order to extend the controller to conduct contact tasks with multiple contact points.

During the programming by demonstration experiments, the recorded position/force trajectory signals that are later used for replay have been downsampled, since the amount of memory at the target interface is. This can become a bigger problem for complete domestic tasks, such as the cutting carrot task, since the amount of data to be recorded will be quite large. Therefore, a compact trajectory learning method which does not require too much memory but captures essentials of the demonstration should be developed.

The identification and the impedance control algorithm should be tested in different kinds of constrained motion tasks such as the ones introduced in chapter 3. It will even be better if these approaches are tested in more complete domestic applications such as the cutting carrot task or similar, introduced in chapter 2. Furthermore, it should be investigated how both of these algorithms can be used together with suitable high level task planning algorithms, so that more complex and complete domestic tasks can be conducted. During this work vision sensing and vision based control has not been used. However it can be realized from the atomic task decomposition introduced in chapter 2 that for more complex tasks vision will also be necessary. Therefore, the integration of vision information together with the identification and control algorithms utilized in this work should further be investigated to create a more complete algorithm.
Bibliography


A. Proof of Stability

In this section the stability and passivity analysis for the Cartesian impedance control law (4.23) – (4.24) for free and constrained motion cases are introduced. The passivity and the stability of the closed loop system given by (4.25) – (4.26) is going to be derived for the case of a constant setpoint (i.e. \( \dot{x}_{\text{ref}} = 0 \), \( \dot{\theta}_{\text{ref}} = 0 \)). The stability analysis which is introduced in this section is restricted to the areas of the workspace where the manipulator configuration is nonsingular, thus the Jacobian, \( J(\theta) \) is also nonsingular. The derivation is first going to be performed for the case of free motion, and then for the case of constrained motion. It can be observed that (4.11) and (4.25) are similar to each other since they both represent the rigid link side dynamics. Therefore the passivity derivation of (4.25), related to the mapping \( \tau \rightarrow \dot{q} \), can be shown by using the same storage function as (4.17) (i.e. \( S_\theta \)). Therefore for free motion case, only the passivity of (4.26) is presented. The passivity of (4.26), related to the mapping \( \dot{q} \rightarrow -\tau \), can be presented with the storage function \( S_\theta \).

\[
S_\theta = \frac{1}{2} \dot{\theta}^T B_\theta \ddot{\theta} + \frac{1}{2} (\theta - q)^T K (\theta - q) + \frac{1}{2} \ddot{x}(\theta) K_x \dddot{x}(\theta) - V_\theta(\theta) \tag{A.1}
\]

where the first term represents the kinetic energy of the rotor dynamics, the second term represents the potential energy stored in the joint flexibilities and the last two terms are the potential energy stored in the controller and gravity potential respectively. When (A.1) is differentiated and (4.26) is used, we obtain

\[
\dot{S}_\theta = \dot{\theta}^T B_\theta \dddot{\theta} + \left( \dot{\theta} - \dot{q} \right)^T K (\theta - q) + \dddot{x}(\theta) K_x \dddot{x}(\theta) - \dot{V}_\theta(\theta)
\]
\[
\dot{S}_\theta = \dot{\theta}^T \left[ \ddot{g}(\theta) - J(\theta)^T (K_x \dddot{x}(\theta) + D_x \dddot{x}) - \tau \right] + (\dot{\theta} - \dot{q})^T \tau + \theta^T J(\theta)^T K_x \dddot{x}(\theta) - \dot{\theta}^T g(\theta)
\]
\[
\dot{S}_\theta = -\dot{\theta}^T J(\theta)^T D_x \dddot{x} - \dot{q}^T \tau = -\dddot{x}^T D_x \dddot{x} - \dot{q}^T \tau \leq -\dot{q}^T \tau \tag{A.2}
\]

Therefore (A.2) satisfies the passivity property given by definition (4.8). The overall closed loop system, (4.11) – (4.12) can be represented as feedback interconnection of the two subsystems, namely the link side and the motor side dynamics, by using their derived passivity properties. The feedback interconnection of these two passive subsystems is shown in Figure A.1.

\[
M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) = \tau
\]
\[
B_\theta \dddot{\theta} + D_x \dddot{x}(\theta) + K_x \left( \Lambda(\theta) - x_{\text{ref}} \right) + \tau = \ddot{g}(\theta)
\]

![Feedback interconnection of passive subsystems](image)

The stability of this closed loop system can be shown with the following candidate Lyapunov function \( V(q, \dot{q}, \theta, \dot{\theta}) \),
\[ V(q, \dot{q}, \theta, \dot{\theta}) = S_q + S_\theta \]  

The derivation for the positive definiteness of this function can be found in [25]. The derivative of (A.3) along the solutions of the closed loop system,

\[ \dot{V} = \dot{S}_q + \dot{S}_\theta = -\dot{x}(\theta)^T D_x \dot{x}(\theta) \]  

is negative semi-definite since \( D_x \) is a positive definite matrix, which shows that the equilibrium point is stable. In order to show that it is asymptotically stable LaSalle’s invariance principle can be used. From the closed-loop dynamics (4.25) – (4.26) and (A.4) it can be observed that there does not exist any trajectory for which \( \dot{x} = 0 \) except the equilibrium \( x(\theta_{eq}) = x_{eq}, \theta = \theta_{eq}, q = q_{eq} \). Asymptotic stability follows from here.

The environment model, (4.32) is used for the passivity and stability analysis in constrained motion. The passivity of (4.25) in the case of constrained motion, related to the mapping \( \tau \to \dot{q} \), can be shown by using the following storage function \( S_q \),

\[ S_q = \frac{1}{2} \dot{q}^T M(q) \dot{q} + V_g(q) + \frac{1}{2} \dot{x}_E^T K_E \dot{x}_E \]  

where \( \dot{x}_E = x(q) - x_E \), and the first term represents the kinetic energy of the rigid body part of the manipulator model and the second term represents the potential energy due to gravity, and the last term stands for the potential energy of the environment modeled as an elasticity. When (A.5) is differentiated and the properties given at the beginning of chapter 4 are used,

\[ \dot{S}_q = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \dot{q}^T (\dot{M}(q) \dot{q} + M(q) \ddot{q}) + \dot{V}_g(q) + \dot{x}_E^T K_E \dot{x}_E \]  

where \( \dot{x}_E = \dot{x}(q) \) since \( x_E \) is assumed to be constant because the interaction between the robot and the environment is modeled as a linear elasticity. The derivation of passivity of (4.26) in the case of constrained motion is similar to the case of free motion since the equations are similar in both cases. The stability of the closed loop system in constrained motion can be derived by using,

\[ V(q, \dot{q}, \theta, \dot{\theta}) = S_q + S_\theta \]  

The derivation for the positive definiteness of this function can be found in [25]. The derivative of (A.7) along the solutions of the closed loop system,

\[ \dot{V} = \dot{S}_q + \dot{S}_\theta = -\dot{x}(\theta)^T D_x \dot{x}(\theta) \]  

is negative semi-definite since \( D_x \) is a positive definite matrix, which shows that the
equilibrium point is stable. From the closed-loop dynamics (4.25) – (4.26) and (A.8) it can be observed that there does not exist any trajectory for which \( \dot{x} = 0 \) except the equilibrium \( x(\theta_{\text{eq}}) = x_{\text{eq}}, \theta = \theta_{\text{eq}}, q = q_{\text{eq}} \). In constrained motion case, the equilibrium position, \( x(\theta_{\text{eq}}) \), of the end-effector is given according to the steady-state analysis presented in section 4.2. When the steady – state friction torques are neglected, it can be observed that the robot will reach the desired setpoint in the unconstrained directions. In the constrained directions, the equilibrium coordinate is determined by the combined effects of environment and manipulator stiffness which can be seen for the two different stiffness matrices selections introduced in section 4.2 as below,

\[
x_{\text{eq}} = mn^T \left[ \frac{k_x}{k_E + k_x} x_{\text{ref}} + \frac{k_E}{k_E + k_x} x_E \right] + (1 - mn^T) x_{\text{ref}} \tag{A.9}
\]

\[
x_{\text{eq}} = mn^T \left[ \frac{k_f}{k_E + k_f} x_{\text{ref}} + \frac{k_E}{k_E + k_f} x_E \right] + (1 - mn^T) x_{\text{ref}} \tag{A.10}
\]
B. Kinematics of the Simulation Setup

B.1. Forward Kinematics

The Denavit – Hartenberg (D-H) parameters which are used in the derivation of the forward kinematics of the simulation model is given in this section [37]. The schematic drawing of the experimental setup is shown in Figure B.1.

![Schematic view of simulation setup](image)

Table B.1 Denavit – Hartenberg parameters

<table>
<thead>
<tr>
<th>Link</th>
<th>a&lt;sub&gt;i&lt;/sub&gt;</th>
<th>α&lt;sub&gt;i&lt;/sub&gt;</th>
<th>d&lt;sub&gt;i&lt;/sub&gt;</th>
<th>q&lt;sub&gt;i&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>-π/2</td>
<td>0</td>
<td>q&lt;sub&gt;1&lt;/sub&gt;*</td>
</tr>
<tr>
<td>2</td>
<td>L</td>
<td>0</td>
<td>l</td>
<td>q&lt;sub&gt;2&lt;/sub&gt;*</td>
</tr>
</tbody>
</table>

By using these D-H parameters, the following homogenous transformation matrices can be written.

\[
A_1 = \begin{bmatrix}
\cos q_1 & 0 & -\sin q_1 & 0 \\
\sin q_1 & 0 & \cos q_1 & 0 \\
0 & -1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\quad A_2 = \begin{bmatrix}
\cos q_2 & -\sin q_2 & 0 & L \cos q_2 \\
\sin q_2 & \cos q_2 & 0 & L \sin q_2 \\
0 & 0 & 1 & l \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

(B.1)

The position and orientation of the tool frame expressed in the base coordinates, is given as,

\[
T_2^0 = A_1 A_2 = \begin{bmatrix}
\cos q_1 \cos q_2 & -\cos q_1 \sin q_2 & -\sin q_1 & L \cos q_1 \cos q_2 - l \sin q_1 \\
\sin q_1 \cos q_2 & -\sin q_1 \sin q_2 & \cos q_1 & L \sin q_1 \cos q_2 + l \cos q_1 \\
-\sin q_2 & -\cos q_2 & 0 & -L \sin q_2 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

(B.2)

B.2. Inverse Kinematics

By using (B.2) from the forward kinematics, the end-effector coordinates can be given as,
\[ x = L \cos q_1 \cos q_2 - l \sin q_i \] (B.3)
\[ y = L \sin q_1 \cos q_2 + l \cos q_i \] (B.4)
\[ z = -L \sin q_2 \] (B.5)

Taking squares of (B.3) and (B.4) and then adding them

\[
x^2 + y^2 = \left( L \cos q_1 \cos q_2 - l \sin q_i \right)^2 + \left( L \sin q_1 \cos q_2 + l \cos q_i \right)^2
\]
\[
x^2 + y^2 = L^2 \cos^2 q_1 \cos^2 q_2 - 2lL \cos q_1 \sin q_i \cos q_2 + l^2 \sin^2 q_i + L^2 \sin^2 q_1 \cos^2 q_2 + 2lL \cos q_1 \sin q_i \cos q_2 + l^2 \cos^2 q_1
\]
\[
x^2 + y^2 = L^2 \cos^2 q_2 + l^2
\] (B.6)

By rearranging (B.6) as,
\[
L \cos q_2 = \pm \sqrt{x^2 + y^2 - l^2}
\] (B.7)

It can be observed from (B.7) that there are 2 solutions for \( q_2 \). If (B.3) and (B.4) are solved for \( \sin q_i \) and \( \cos q_i \),

\[
\begin{bmatrix}
x \\
y
\end{bmatrix} =
\begin{bmatrix}
-l & L \cos q_2 \\
L \cos q_2 & l
\end{bmatrix}^{-1}
\begin{bmatrix}
\sin q_i \\
\cos q_i
\end{bmatrix}
\]

\[
\begin{bmatrix}
x \\
y
\end{bmatrix} =
\begin{bmatrix}
1 & -L \cos q_2 \\
L \cos q_2 & -l
\end{bmatrix}^{-1}
\begin{bmatrix}
x \\
y
\end{bmatrix}
\begin{bmatrix}
\sin q_i \\
\cos q_i
\end{bmatrix}
\]

\[
\frac{1}{l^2 + L^2 \cos^2 q_2} \begin{bmatrix}
-l & L \cos q_2 \\
L \cos q_2 & l
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
\begin{bmatrix}
\sin q_i \\
\cos q_i
\end{bmatrix}
\]

\[
\sin q_1 = \frac{-xl + yL \cos q_2}{l^2 + L^2 \cos^2 q_2}
\]
\[
\cos q_1 = \frac{xL \cos q_2 + yl}{l^2 + L^2 \cos^2 q_2}
\] (B.8)

By using (B.6), (B.7) and (B.8), two solutions for \( q_1 \) can be found as a function of the end-effector coordinates \((x,y,z)\) as follows,

\[
\sin q_1 = \frac{-xl + y \sqrt{x^2 + y^2 - l^2}}{x^2 + y^2}
\]
\[
\cos q_1 = \frac{x \sqrt{x^2 + y^2 - l^2} + yl}{x^2 + y^2}
\]

**Solution 1:** \( q_i = \tan^{-1} \left( \frac{-xl + y \sqrt{x^2 + y^2 - l^2}}{x \sqrt{x^2 + y^2 - l^2} + yl} \right) \)

\[
\sin q_i = \frac{-xl - y \sqrt{x^2 + y^2 - l^2}}{x^2 + y^2}
\]
\[
\cos q_i = \frac{-x \sqrt{x^2 + y^2 - l^2} + yl}{x^2 + y^2}
\]
solution 2: \( q_1 = \tan^{-1}\left(\frac{-x\sqrt{x^2 + y^2 - L^2} + y\sqrt{x^2 + y^2}}{x^2 + y^2}\right) \) \hspace{1cm} (B.9)

By using (B.5) and (B.7), two solutions for \( q_2 \) can be found as a function of the end-effector coordinates \((x,y,z)\) as follows,

\[
\begin{align*}
\sin q_2 &= \frac{-z}{L} \\
\cos q_2 &= \frac{\sqrt{x^2 + y^2 - L^2}}{L} \\
\end{align*}
\]

solution 1: \( q_2 = \tan^{-1}\left(\frac{-z\sqrt{x^2 + y^2 - L^2}}{L}\right) \)

\[
\begin{align*}
\sin q_2 &= \frac{-z}{L} \\
\cos q_2 &= -\frac{\sqrt{x^2 + y^2 - L^2}}{L} \\
\end{align*}
\]

solution 2: \( q_2 = \tan^{-1}\left(\frac{-z\sqrt{x^2 + y^2 - L^2}}{L}\right) \) \hspace{1cm} (B.10)

B.3. Jacobian

The derivation of the manipulator Jacobian matrix for the simulation model is given in this section [37].

\[
J = \begin{bmatrix}
J_{v1} & J_{v2} \\
J_{o1} & J_{o2}
\end{bmatrix} = \begin{bmatrix}
z_0 \times (o_2 - o_0) & z_0 \times (o_2 - o_1) \\
z_0 & z_1
\end{bmatrix}
\] \hspace{1cm} (B.11)

\[
z_0 = \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix} \quad \begin{bmatrix}
\sin q_1 \\
\cos q_1 \\
0
\end{bmatrix} \quad o_0 = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix} \quad o_2 = \begin{bmatrix}
L \cos q_1 \cos q_2 - l \sin q_1 \\
L \sin q_1 \cos q_2 + l \cos q_1 \\
-L \sin q_2
\end{bmatrix}
\]

\[
z_0 \times (o_2 - o_0) = \begin{bmatrix}
-L \sin q_1 \cos q_2 - l \cos q_1 \\
L \cos q_1 \cos q_2 - l \sin q_1 \\
0
\end{bmatrix} \quad z_1 \times (o_2 - o_1) = \begin{bmatrix}
-L \cos q_1 \sin q_2 \\
-L \sin q_1 \sin q_2 \\
-L \cos q_2
\end{bmatrix}
\] \hspace{1cm} (B.12)

By using (B.11), (B.12) and (B.13) the manipulator Jacobian becomes as follows,

\[
J = \begin{bmatrix}
-L \sin q_1 \cos q_2 - l \cos q_1 & -L \cos q_1 \sin q_2 \\
L \cos q_1 \cos q_2 - l \sin q_1 & -L \sin q_1 \sin q_2 \\
0 & -L \cos q_2 \\
0 & -\sin q_1 \\
0 & \cos q_1 \\
1 & 0
\end{bmatrix}
\] \hspace{1cm} (B.14)
B.4. Dynamic Model

The dynamic model of the manipulator used during the simulations is given by (4.1), (4.2), and (4.3). The manipulator which is used during the simulations contains 2 degrees-of-freedom, therefore the entry given by \( n = 2 \) and \( q = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^T \). The inertia tensor of each link are given as,

\[
I_1 = \begin{bmatrix} I_{1xx} & I_{1xy} & I_{1xz} \\ I_{1yx} & I_{1yy} & I_{1yx} \\ I_{1zx} & I_{1zy} & I_{1zz} \end{bmatrix} \quad I_2 = \begin{bmatrix} I_{2xx} & I_{2xy} & I_{2xz} \\ I_{2yx} & I_{2yy} & I_{2yx} \\ I_{2zx} & I_{2zy} & I_{2zz} \end{bmatrix} \tag{B.15}
\]

The elements of the configuration dependent inertia matrix, \( M(q) \) are given as,

\[
M(q) = \begin{bmatrix} m_{11}(q) & m_{12}(q) \\ m_{21}(q) & m_{22}(q) \end{bmatrix}
\]

\[
m_{11}(q) = m_1 I_{1xx} + m_2 L^2 \cos^2(q_2) + m_2 I_{2xx} \cos^2(q_2) + m_2 I_{2xy}^2 + I_{2yy} \cos(q_2)^2 + I_{2xz} + I_{1xy} \\
-2m_2 I_{2xy} \cos(q_2) + m_2 I_{2xy} - I_{2xx} \cos^2(q_2) - 2m_2 I_{2xx} L \cos^2(q_2) + 2m_2 I_{2xy} L \cos(q_2) \sin(q_2) \\
-2m_2 I_{2xy} \cos(q_2) \sin(q_2) + I_{2yy} \sin(q_2) \cos(q_2) + I_{2xy} \cos(q_2) \sin(q_2)
\]

\[
m_{12}(q) = -I_{2yy} \cos(q_2) - I_{2xx} \sin(q_2) + m_2 L \cdot L \sin(q_2) - m_2 L \cdot I_{2xx} \sin(q_2) - m_2 L \cdot I_{2xy} \cos(q_2) - I_{2xy} \sin(q_2) - I_{2yy} \cos(q_2)
\]

\[
m_{21}(q) = m_2 L \cdot \sin(q_2) - m_2 I_{2xx} \sin(q_2) - m_2 I_{2xy} \cos(q_2) - I_{2xx} \sin(q_2) - I_{2xy} \cos(q_2)
\]

\[
m_{22}(q) = -2m_2 I_{2xy} L + m_2 L^2 + m_2 I_{2xx}^2 + m_2 I_{2xy}^2 + I_{2xx}
\]

The Christoffel symbols which are used to calculate the elements of the Coriolis and centrifugal terms are provided as follows,

\[
c_{111}(q) = 0
\]

\[
c_{112}(q) = \frac{1}{2} m_2 L^2 \sin(2q_2) + \frac{1}{2} m_2 I_{2xx}^2 \sin(2q_2) + \frac{1}{2} I_{2yy} \sin(2q_2) - \frac{1}{2} m_2 I_{2xy} \sin(2q_2)
\]

\[
-\frac{1}{2} I_{2xx} \sin(2q_2) - m_2 I_{2xx} L \sin(2q_2) - m_2 I_{2xy} L \cos(2q_2) + m_2 I_{2xx}^2 I_{2xy} \cos(2q_2)
\]

\[
-\frac{1}{2} I_{2yy} \cos(2q_2) - \frac{1}{2} I_{2xy} \cos(2q_2)
\]

\[
c_{121}(q) = -\frac{1}{2} m_2 L^2 \sin(2q_2) - \frac{1}{2} m_2 I_{2xx}^2 \sin(2q_2) - \frac{1}{2} I_{2yy} \sin(2q_2) + \frac{1}{2} m_2 I_{2xy} \sin(2q_2)
\]

\[
+\frac{1}{2} I_{2xx} \sin(2q_2) + m_2 I_{2xx} L \sin(2q_2) + m_2 I_{2xy} L \cos(2q_2) - m_2 I_{2xx}^2 I_{2xy} \cos(2q_2)
\]

\[
+\frac{1}{2} I_{2yy} \cos(2q_2) + \frac{1}{2} I_{2xy} \cos(2q_2)
\]

\[
c_{122}(q) = -\frac{1}{2} I_{2yy} \cos(q_2) + \frac{1}{2} I_{2yy} \sin(q_2) - \frac{1}{2} I_{2yy} \sin(q_2) + \frac{1}{2} I_{2xy} \cos(q_2)
\]

\[
c_{121}(q) = -\frac{1}{2} m_2 I_{2xx} \sin(2q_2) - \frac{1}{2} I_{2xy} \sin(2q_2) + \frac{1}{2} m_2 I_{2xy} \sin(2q_2)
\]

\[
+\frac{1}{2} I_{2xx} \sin(2q_2) + m_2 I_{2xx} L \sin(2q_2) + m_2 I_{2xy} L \cos(2q_2) - m_2 I_{2xx}^2 I_{2xy} \cos(2q_2)
\]

\[
+\frac{1}{2} I_{2yy} \cos(2q_2) + \frac{1}{2} I_{2xy} \cos(2q_2)
\]

\[
c_{122}(q) = -\frac{1}{2} I_{2yy} \cos(q_2) + \frac{1}{2} I_{2yy} \sin(q_2) - \frac{1}{2} I_{2yy} \sin(q_2) + \frac{1}{2} I_{2xy} \cos(q_2)
\]
\begin{align*}
c_{211}(q) &= -\frac{1}{2}m_{2}L_{x}^{2}\sin(2q_{2}) - \frac{1}{2}m_{2}L_{z}^{2}\sin(2q_{2}) - \frac{1}{2}I_{2xy}^{2}\sin(2q_{2}) + \frac{1}{2}m_{2}L_{y}^{2}\sin(2q_{2}) \\
&\quad + \frac{1}{2}I_{2xz}\sin(2q_{2}) + m_{2}L_{xz}L_{x}\sin(2q_{2}) + m_{2}L_{yz}\cos(2q_{2}) - m_{2}L_{xz}L_{z}\cos(2q_{2}) \\
&\quad + \frac{1}{2}I_{2xy}\cos(2q_{2}) + \frac{1}{2}I_{2yz}\cos(2q_{2}) \\
c_{212}(q) &= 0 \\
c_{221}(q) &= I_{2xy}\sin(q_{2}) - I_{2xz}\cos(q_{2}) + m_{2}L\cos(q_{2}) - m_{2}L_{xz}\cos(q_{2}) + m_{2}L_{yz}\sin(q_{2}) \\
c_{222}(q) &= 0
\end{align*}

(B.17)

By using these Christoffel symbols, the Coriolis and centrifugal terms, \( C(q,\dot{q})\dot{q} \) can be calculated as,

\[
C(q,\dot{q})\dot{q} = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ijk} (q) \dot{q}_{i} \dot{q}_{j}
\]  

(B.18)

The elements of the gravity torques vector, \( g(q) \) are given as,

\[
g_{11}(q) = -m_{1}g_{1z}\sin(q_{1}) + m_{2}g\left[ L_{xz}\cos(q_{1}) - L_{xy}\sin(q_{1}) - l_{2xz}\cos(q_{1}) \cos(q_{2}) + l_{2xy}\cos(q_{1}) \sin(q_{2}) \right]
\]

\[
g_{21}(q) = -m_{2}g \cdot \sin(q_{1}) \left[ L_{x}\sin(q_{2}) - l_{2xz}\sin(q_{2}) - l_{2xy}\cos(q_{2}) \right]
\]  

(B.19)

\begin{table}[h]
\centering
\caption{Parameters used in the simulation model}
\begin{tabular}{|c|c|c|c|c|}
\hline
\( l \) & 0.05 m & \( L \) & 0.7 m & \( l_{1z} \) & 0.025 m & \( a \) & 0.803 m \\
\hline
\( l_{2cx} \) & 0.4667 m & \( l_{2cy} \) & 0.803 m & \( m_{1} \) & 0.6 kg & \( m_{2} \) & 2 kg \\
\hline
\( I_{jxx} \) & 0.00035 kgm\(^{2}\) & \( I_{jyy} \) & 0.00035 kgm\(^{2}\) & \( I_{1zz} \) & 0.0006375 kgm\(^{2}\) & \( I_{1xy} \) & 0 kgm\(^{2}\) \\
\hline
\( I_{jyx} \) & 0 kgm\(^{2}\) & \( I_{1yz} \) & 0 kgm\(^{2}\) & \( I_{1xz} \) & 0 kgm\(^{2}\) & \\
\hline
\( I_{2yz} \) & 0 kgm\(^{2}\) & \( I_{2yz} \) & 0.0031875 kgm\(^{2}\) & \( I_{2zz} \) & 0.0019125 kgm\(^{2}\) & \( I_{2yz} \) & 0.00105 kgm\(^{2}\) \\
\hline
\( I_{2xy} \) & 0 kgm\(^{2}\) & \( I_{2xy} \) & 0 kgm\(^{2}\) & \( I_{2xz} \) & 0 kgm\(^{2}\) & \( I_{2z} \) & 0 kgm\(^{2}\) \\
\hline
\( I_{2ze} \) & 0 kgm\(^{2}\) & \( I_{2z} \) & 0 kgm\(^{2}\) & \( B \) & 0.1592 kgm\(^{2}\) & \( K \) & 5000 N/m \\
\hline
\end{tabular}
\end{table}
C. Kinematics of the Experimental Setup

C.1. Forward Kinematics

The Denavit – Hartenberg (D-H) parameters which are used in the derivation of the forward kinematics of the experimental setup is given in this section [37]. The schematic drawing of the experimental setup is shown in Figure C.1.

![Figure C.1 Front schematic view and Left schematic view](image)

The Denavit – Hartenberg (D-H) parameters which are used in the derivation of the forward kinematics of the experimental setup is given in this section [37].

<table>
<thead>
<tr>
<th>Link</th>
<th>a_i</th>
<th>α_i</th>
<th>d_i</th>
<th>q_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>-π/2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-π/2</td>
<td>0</td>
<td>q_1^*</td>
</tr>
<tr>
<td>3</td>
<td>L_2</td>
<td>0</td>
<td>l</td>
<td>q_2^*</td>
</tr>
<tr>
<td>4</td>
<td>L_3</td>
<td>0</td>
<td>0</td>
<td>π/2 − β</td>
</tr>
</tbody>
</table>

Here the parameters given in Table C.1 and Figure C.1 are \( l = 0.085 \) m, \( L_1 = 0.4 \) m, \( L_2 = 0.27 \) m, \( L_3 = 0.093 \) m, \( L = 0.5525 \) m, \( n = 0.463 \) m, \( r = 0.5455 \) m, \( β = 0.3416 \) rad,
\[ \gamma = 0.4785 \text{ rad}, \ \delta = 0.4532 \text{ rad}. \] By using these D-H parameters and after rearranging some terms, the end-effector coordinates can be written as follows.

\[ \begin{align*}
    x &= (a \sin q_2 + b \cos q_2) \cos q_1 - l \sin q_1 \\
    y &= -b \sin q_2 + a \cos q_2 \\
    z &= (-a \sin q_2 - b \cos q_2) \sin q_1 - l \cos q_1
\end{align*} \] (C.1) (C.2) (C.3)

where \( a = L_3 \sin \beta - L_2 \cos \beta, b = L_3 \cos \beta + L_2 \sin \beta + L_1. \)

**C.2. Inverse Kinematics**

By using the end-effector coordinates given in the previous section (C.1) and (C.3) can be written as,

\[
\begin{bmatrix}
    x \\
    z
\end{bmatrix} = \begin{bmatrix}
    -l & m \\
    -m & -l
\end{bmatrix} \begin{bmatrix}
    \sin q_1 \\
    \cos q_1
\end{bmatrix} \quad \text{and} \quad \frac{1}{m^2 + l^2} \begin{bmatrix}
    -l & -m \\
    m & -l
\end{bmatrix} \begin{bmatrix}
    x \\
    z
\end{bmatrix} = \begin{bmatrix}
    \sin q_1 \\
    \cos q_1
\end{bmatrix}
\] (C.4)

where \( m = a \sin q_2 + b \cos q_2. \) When this is used together with (C.2) as,

\[ m^2 + y^2 = a^2 + b^2 \] (C.5)

By summing the squares of (C.1), (C.2) and (C.3) and then rearranging them by using (C.5),

\[
\begin{align*}
    x^2 + y^2 + z^2 &= a^2 + b^2 + l^2 \\
    x^2 + z^2 &= l^2 + m^2 \\
    m &= \sqrt{x^2 + z^2 - l^2}
\end{align*}
\] (C.6)

By using (C.4) and (C.6), two solutions for \( q_1 \) can be found as a function of the end-effector coordinates \((x, y, z)\) as follows,

\[
\begin{align*}
    \sin q_1 &= \frac{-xl - z\sqrt{x^2 + z^2 - l^2}}{m^2 + l^2}, & \cos q_1 &= \frac{x\sqrt{x^2 + z^2 - l^2} - lz}{m^2 + l^2} \\
    \text{solution 1: } q_1 &= \arctan 2 \left( \frac{-xl - z\sqrt{x^2 + z^2 - l^2}}{x^2 + z^2}, \frac{x\sqrt{x^2 + z^2 - l^2} - lz}{x^2 + z^2} \right) \\
    \sin q_1 &= \frac{-xl + z\sqrt{x^2 + z^2 - l^2}}{m^2 + l^2}, & \cos q_1 &= \frac{-x\sqrt{x^2 + z^2 - l^2} - lz}{m^2 + l^2} \\
    \text{solution 2: } q_1 &= \arctan 2 \left( \frac{-xl + z\sqrt{x^2 + z^2 - l^2}}{x^2 + z^2}, \frac{-x\sqrt{x^2 + z^2 - l^2} - lz}{x^2 + z^2} \right)
\end{align*}
\] (C.7)

By using (C.2) and (C.6), two solutions for \( q_2 \) can be found as a function of the end-effector coordinates \((x, y, z)\) as follows,

\[
\begin{bmatrix}
    m \\
    y
\end{bmatrix} = \begin{bmatrix}
    a & b \\
    -b & -a
\end{bmatrix} \begin{bmatrix}
    \sin q_2 \\
    \cos q_2
\end{bmatrix} \quad \text{and} \quad \frac{1}{a^2 + b^2} \begin{bmatrix}
    a & -b \\
    b & a
\end{bmatrix} \begin{bmatrix}
    m \\
    y
\end{bmatrix} = \begin{bmatrix}
    \sin q_2 \\
    \cos q_2
\end{bmatrix}
\] (C.8)
\[
\sin q_2 = \frac{a\sqrt{x^2 + z^2 - l^2} - by}{a^2 + b^2}, \quad \cos q_2 = \frac{b\sqrt{x^2 + z^2 - l^2} + ay}{a^2 + b^2}
\]

**solution 1:** \(q_2 = \text{atan} \left( \frac{a\sqrt{x^2 + z^2 - l^2} - by}{b\sqrt{x^2 + z^2 - l^2} + ay} \right) \)

\[
\sin q_2 = \frac{-a\sqrt{x^2 + z^2 - l^2} - by}{a^2 + b^2}, \quad \cos q_2 = \frac{-b\sqrt{x^2 + z^2 - l^2} + ay}{a^2 + b^2}
\]

**solution 2:** \(q_2 = \text{atan} \left( \frac{-a\sqrt{x^2 + z^2 - l^2} - by}{-b\sqrt{x^2 + z^2 - l^2} + ay} \right) \) \hspace{1cm} (C.9)

### C.3. Jacobian

The Jacobian matrix of the manipulator used in the experimental setup is given in this section.

\[
J = \begin{bmatrix}
-(a \sin q_2 + b \cos q_2) \sin q_i - l \cos q_i & (a \cos q_2 - b \sin q_2) \cos q_i \\
0 & -b \cos q_2 - a \sin q_2 \\
-(a \sin q_2 + b \cos q_2) \cos q_i + l \sin q_i & (-a \cos q_2 + b \sin q_2) \sin q_i
\end{bmatrix}
\] \hspace{1cm} (C.10)

### C.4. Gravity Torques

The masses of each separate part as shown in Figure C.1 are \(m_1 = 0.33 \text{ kg}, m_2 = 0.225 \text{ kg}, m_3 = 0.59 \text{ kg}, m_4 = 0.3 \text{ kg}\). The gravity torques are given as follows,

<table>
<thead>
<tr>
<th>Link</th>
<th>(a_i)</th>
<th>(\alpha_i)</th>
<th>(d_i)</th>
<th>(q_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>(-\pi / 2)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c.o.g.1</td>
<td>(L_1 / 2)</td>
<td>0</td>
<td>(l)</td>
<td>(q_2^*)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Link</th>
<th>(a_i)</th>
<th>(\alpha_i)</th>
<th>(d_i)</th>
<th>(q_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>(-\pi / 2)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c.o.g.2</td>
<td>(n)</td>
<td>0</td>
<td>(l)</td>
<td>(q_2^* + \gamma / 2)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Link</th>
<th>(a_i)</th>
<th>(\alpha_i)</th>
<th>(d_i)</th>
<th>(q_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>(-\pi / 2)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c.o.g.3</td>
<td>(r)</td>
<td>0</td>
<td>(l)</td>
<td>(q_3^* + \delta)</td>
</tr>
</tbody>
</table>
The coordinates of the center of masses of each separate part are given as follows,

\begin{align*}
\mathbf{r}_{\text{cog},1} &= \begin{bmatrix}
\frac{L_1}{2} \cos q_1 \cos q_2 - l \sin q_1 \\
\frac{L_1}{2} \sin q_1 \\
\frac{L_1}{2} \sin q_1 \cos q_2 - l \cos q_1
\end{bmatrix}^T \\
\mathbf{r}_{\text{cog},2} &= \begin{bmatrix}
\cos q_1 \cos \left( q_2 + \frac{\gamma}{2} \right) - l \sin q_1 \\
-\sin \left( q_2 + \frac{\gamma}{2} \right) - n \sin q_1 \cos \left( q_2 + \frac{\gamma}{2} \right) - l \cos q_1
\end{bmatrix}^T \\
\mathbf{r}_{\text{cog},3} &= \begin{bmatrix}
r \cos q_1 \cos (q_2 + \delta) - l \sin q_1 \\
-r \sin (q_2 + \delta) - r \sin q_1 \cos (q_2 + \delta) - l \cos q_1
\end{bmatrix}^T \\
\mathbf{r}_{\text{cog},4} &= \begin{bmatrix}
\frac{l}{2} \cos q_1 \\
0 \\
-\frac{l}{2} \sin q_1
\end{bmatrix}^T
\end{align*}

The total gravitational potential energy of the robot can be written as,

\begin{align*}
V_g(q) &= m_1 g^T \mathbf{r}_{\text{cog},1} + m_2 g^T \mathbf{r}_{\text{cog},2} + m_3 g^T \mathbf{r}_{\text{cog},3} + m_4 g^T \mathbf{r}_{\text{cog},4} \\
&= -m_1 g \left( \frac{L_1}{2} \sin q_1 \cos q_2 - l \cos q_1 \right) - m_2 g \left(-n \sin q_1 \cos \left( q_2 + \frac{\gamma}{2} \right) - l \cos q_1 \right) \\
&- m_3 g \left(-r \sin q_1 \cos (q_2 + \delta) - l \cos q_1 \right) - m_4 g \left( -\frac{l}{2} \sin q_1 \right) \\
&= \left( m_1 g \frac{L_1}{2} \cos q_2 + m_2 g n \cos \left( q_2 + \frac{\gamma}{2} \right) + m_3 g r \cos (q_2 + \delta) + m_4 g \frac{l}{2} \right) \sin q_1 \\
&\quad + (m_1 + m_2 + m_3) g l \cos q_1
\end{align*}

(C.12)

From \(V_g(q)\), the gravity torque vector can be calculated by,

\begin{align*}
g(q) &= \frac{\partial V_g(q)}{\partial q} = \begin{bmatrix}
g_{11}(q) \\
g_{21}(q)
\end{bmatrix} \\
g_{11}(q) &= \begin{bmatrix}
m_1 g \frac{L_1}{2} \cos q_2 + m_2 g n \cos \left( q_2 + \frac{\gamma}{2} \right) + m_3 g r \cos (q_2 + \delta) + m_4 g \frac{l}{2}
\end{bmatrix} \cos q_1 \\
&\quad -(m_1 + m_2 + m_3) g l \sin q_1 \\
g_{21}(q) &= \begin{bmatrix}
m_1 g \frac{L_1}{2} \sin q_2 + m_2 g n \sin \left(q_2 + \frac{\gamma}{2}\right) + m_3 g r \sin (q_2 + \delta)
\end{bmatrix} \sin q_1
\end{align*}

(C.13)
D. Calibration

D.1. ATI 6 d.o.f. Force/Torque Sensor

The contact impedance identification method introduced in chapter 3 and implemented in experiments in chapter 6, makes use of a 6 d.o.f force/torque sensor from ATI. It is mounted at the end-effector side of the second link (i.e. the curved link). A stiff pin-like object is also mounted at the force sensor in order to achieve a point contact between the end-effector and the environment. The force measurements using this sensor are affected by the weight of this object and the small plate holding this object. Therefore, these should be removed from the force sensor readings. The calibration block diagram of the force sensor which is used during experiments is displayed in Figure D.1 as follows.

![Figure D.1 6 d.o.f. F/T sensor calibration diagram](image)

The end-effector orientation is calculated by using the forward kinematics of the robot. The calibration matrix in Figure D.1 is obtained from ATI force sensor manual [39]. It represents the relation between the strain gage voltages and forces/torques. The weight of the stiff-pin like object together with the small plate is 0.95 N.

D.2. Joint Torque Sensors

During the impedance control experiments in chapter 6, the measured joint torques are also used in the control laws. The joint torque sensors are calibrated by putting the arm in a stiff close loop control and then hanging several different weights. Then checking the effect of these weights. The calibration data for both torque sensors are given in Table D.1.

<table>
<thead>
<tr>
<th>Weight</th>
<th>T1 [Nm]</th>
<th>V1 [V]</th>
<th>T2 [Nm]</th>
<th>V2 [V]</th>
</tr>
</thead>
<tbody>
<tr>
<td>no extra weight</td>
<td>2.0947</td>
<td>2.437</td>
<td>2.072</td>
<td>2.509</td>
</tr>
<tr>
<td>0.05 kg extra weight</td>
<td>1.4644</td>
<td>2.441</td>
<td>1.4417</td>
<td>2.506</td>
</tr>
<tr>
<td>0.1 kg extra weight</td>
<td>1.0862</td>
<td>2.445</td>
<td>1.0635</td>
<td>2.503</td>
</tr>
<tr>
<td>0.2 kg extra weight</td>
<td>0.9602</td>
<td>2.453</td>
<td>0.9375</td>
<td>2.497</td>
</tr>
<tr>
<td>0.5 kg extra weight</td>
<td>0.8971</td>
<td>2.474</td>
<td>0.8744</td>
<td>2.479</td>
</tr>
<tr>
<td>1 kg extra weight</td>
<td>0.8341</td>
<td>2.515</td>
<td>0.8114</td>
<td>2.451</td>
</tr>
</tbody>
</table>
The relationship between the measured ADC channel voltages and the change in weights is used to determine the gain and the bias of the torque sensors. A property of these sensors is that the torque is inversely proportional to the voltage output. The calibration data and fitting them with linear approximations are given in Figures D.2 and D.3.

Figure D.2  Calibration data and linear fit for 1st joint torque sensor

Figure D.3  Calibration data and linear fit for 2nd joint torque sensor

The gain and bias are determined from these linear approximations given in Figures D.2 and D.3. The joint torque signals are also filtered with a weighted moving average filter to reduce the measurement noise.
The properties of the DC motor of the experimental setup which are given in Table 6.1 are obtained from the catalog given in Figure E.1. The specifications of the DC motor with the order number 310009 are used.

**E. Maxon Motor Catalog**

The properties of the DC motor of the experimental setup which are given in Table 6.1 are obtained from the catalog given in Figure E.1. The specifications of the DC motor with the order number 310009 are used.

---

**Figure E.1** Maxon motor catalog
**F. Additional Results**

In this section the additional experimental results and additional figures related to the simulation results from chapter 5 and experimental results from chapter 6 are presented. Furthermore, an approach to determine the contact normal direction from the estimated stiffness matrix is also presented together with some supporting results.

**F.1. Additional Simulation Results**

The mass matrix which is related to the identification simulation introduced in section 5.1 is given here below in Figure F.1.

![Graph showing mass terms](image)

**Figure F.1** Estimated mass terms by recursive/batch algorithms and corresponding uncertainties

It can be observed from figure F.1 that the estimated mass terms during contact are close to zero, since no mass related to the environment has been modeled in the simulations.
F.2. Additional Experimental Results

The mass matrix terms that are obtained by the impedance estimation algorithm related to the pressing task in section 6.2 are given in Figure F.2.

![Figure F.2 Estimated mass terms by recursive/batch algorithms and corresponding uncertainties](image)

The estimated mass matrix terms are mostly zero in constrained motion phase, most probably because of two reasons: 1) the signals are not sufficiently persistently exciting to estimate the higher order terms, such as the mass matrix and 2) the bilinear transformation is not accurate enough to approximate the accelerations.

Next, the additional results related to the sliding experiment given in section 6.2 are presented. The norm of the end-effector velocity is given in Figure F.3.

![Figure F.3 Norm of the end-effector velocity](image)
According to the selection of the forgetting factor parameters, the bound on the low-speed motion becomes $X_{hi}/T_{hi} = 0.02 \text{ m/s}$. The bias terms that are estimated using the impedance perception algorithm are given in Figure F.4.

The straight lines in these figures represent the estimated components of the impedance determined using the recursive algorithm, the horizontal dash-dot lines are the estimates achieved using the batch least squares algorithm, the gray bands represent the uncertainties of the estimates, and the vertical dotted lines indicate the instants when the discontinuities are detected (i.e., gain and loss of contact in this case). The batch processing is applied when the end-effector is in constrained motion phase. The liftboy surface’s $z$ coordinate at the contact point is approximately at $-0.4586 \text{ m}$, which can be observed in Figure 6.10. The absolute value of this coordinate can also be calculated based on the bias and stiffness values (constrained motion phase) obtained using the batch processing as follows

$$\frac{c_z}{K_{zz}} = \frac{307.2}{673.5} \approx 0.4561 [\text{m}]$$

which is close to the measured value. The damping terms that are obtained by the impedance estimation algorithm related to the sliding task in section 6.2 are given in Figure F.5. It can be observed that the uncertainties of the damping terms are relatively higher, which is most probably due to the fact that the input signals are not sufficiently persistently exciting.
The mass matrix terms that are obtained by the impedance estimation algorithm are given in Figure F.6. The estimated mass matrix terms are mostly zero, the most probably because of two reasons: 1) the signals are not sufficiently persistently exciting to estimate the higher order terms, such as the mass matrix and 2) the bilinear transformation is not accurate enough to approximate the accelerations.

The third impedance identification experiment which is given here as mentioned in section 6.2, is related to pressing on a compliant (soft) object. The demonstrated task again consists of the same three phases as in the case of pressing on the stiff object. The end-effector positions and forces recorded during the demonstrations are shown in Figure F.7 and F.8 respectively. It can be observed that the force in the z-direction displays a significant change compared to the
forces in x-direction while the tip is in contact. The z-coordinate of the end-effector is also not nearly constant while it is in contact, which was the case for the stiff object. The sponge deforms approximately 45 mm in the z direction and 2 mm in the x direction which can be observed from Figure F.8.

![Figure F.7 End-effector forces for pressing on a soft object](image)

![Figure F.8 End-effector positions for pressing on a soft object](image)

The parameters of the virtual soft finger (V.S.F.) that are used during the experiments are \( K_v = 700 \, \text{N/m} \) and \( B_v = 10 \, \text{Ns/m} \). The parameters of the forgetting factor are \( X_H = 0.02 \, \text{m} \) and \( T_H = 1 \, \text{sec} \). The uncertainty estimation parameter is \( A = 0.5 \). The norm of the end-effector velocity is given in Figure F.9. According to the selection of the forgetting factor parameters, the bound on the low-speed motion becomes \( X_H / T_H = 0.02 \, \text{m/s} \).
The discontinuities (i.e. gain and loss of contact) are detected using the thresholding on the norm of the force, instead of using the second method mentioned in section 3.3.4 for the same reason as in the case of pressing on the stiff object. The norm of the end-effector forces is given in Figure F.10. The threshold on the force is selected as 0.3 [N].

The estimation results obtained from the impedance perception algorithm are given in Figures F.11, F.12, F.13 and F.14. The straight lines in these figures represent the estimated components of the impedance determined using the recursive algorithm, the horizontal dash-dot lines are the estimates achieved using the batch least squares algorithm, the gray bands represent the uncertainties of the estimates, and the vertical dotted lines indicate the instants when the discontinuities are detected (i.e. gain and loss of contact in this case). The bias terms that are estimated using the impedance perception algorithm are given in Figure F.11.
The estimated discontinuity instants are close to the actual moments when the discontinuities occur. This is confirmed by the end-effector force plots shown in Figure F.7, since at 10 second the end-effector forces become different than zero and start increasing, while at 42.3rd second the end-effector forces become zero again. The stiffness coefficients that are obtained using the impedance estimation algorithm are given in Figure F.12.

It can be observed that when the end-effector is in contact with the sponge, the stiffness $K_{zz}$, is equal to 118 N/m with a small uncertainty. The stiffness in the tangential direction component $K_{xx}$, when the end-effector is in contact with the sponge, fluctuates with a relatively larger uncertainty. It can also be observed that the off diagonal term $K_{zx}$ converges to zero with a small uncertainty, while the other $K_{xz}$ fluctuates with a relatively larger uncertainty, the most probably due to the nonlinear behavior of the sponge. Both the bias and
the stiffness values during two unconstrained motion phases are close to zero with a low uncertainty, if we exclude the short transient effects. The $z$ coordinate on the surface, at the moment (10 sec) when the end-effector makes the first contact with the surface, is approximately -0.5 m, which can be observed in Figure F.8. The absolute value of this coordinate can be calculated as is done before, by using the bias and stiffness coefficients (from the constrained motion phase) estimated using the batch processing:

$$
\frac{c_z}{K_{zz}} = \frac{54.56}{117.8} \approx 0.4631 \text{[m]}
$$

This value is not that close to the measured coordinate in comparison with the two previous experiments. This indicates that the linear contact model is not sufficient to describe the behavior of the sponge.

The damping terms that are obtained by the impedance estimation algorithm related to the pressing on sponge task in section 6.2 are given in Figure F.13. It can be observed that the uncertainties of the damping terms are relatively higher, which is the most probably due to the fact that the input signals are not sufficiently persistently exciting, and also because of the nonlinear behavior of the sponge.

The mass matrix terms that are obtained by the impedance estimation algorithm are given in Figure F.14. The estimated mass matrix terms are mostly zero, the most probably because of two reasons: 1) the signals are not sufficiently persistently exciting to estimate the higher order terms, such as the mass matrix and 2) the bilinear transformation is not accurate enough to approximate the accelerations.
The estimator’s performance can also be checked by comparing the output (i.e. forces) of the model given by

$$\hat{\phi}_k = \Theta^T \psi_k$$  \hspace{1cm} (F.1)

with the measured (raw) outputs. The measured outputs, the estimated outputs and their difference are given in Figure F.15. It can be observed that the estimator’s outputs for the 2nd element of $\hat{\phi}_k$ (i.e. force in z direction) matches the sensed signal reasonably (except from the discontinuity at 42.3 second), whereas for the 1st element this error is relatively higher. This is most probably because of the nonlinear features of the sponge.
Next, the experiment which is related to pressing down on a stiff surface mentioned in section 6.3.2 is given here. At the beginning of the experiment, the end-effector is brought on top of the stiff surface in zero-gravity mode. Then, the Cartesian control mode is activated to execute the experiment. During the experiment only two directions, namely x and z, of the Cartesian control law are used together with their corresponding reference trajectories (i.e. position/force). The end-effector is commanded to exert 10 N by moving 0.1 m through the object, without moving sideways. The end-effector forces and positions recorded during the experiment are shown in Figure F.16 and F.17, respectively. The straight lines in Figure F.16 are related to the end-effector forces and the dashed lines are the reference trajectory signals. The straight lines in Figure F.17 are related to the end-effector positions calculated from the forward kinematics based on encoder signals, and the dashed lines are the reference trajectory signals.

![Figure F.16](image_url) End-effector forces for pressing on a stiff object

![Figure F.17](image_url) End-effector positions for pressing on a stiff object
The controller parameters that are used during this experiment are $K_{xx} = 3000 \text{ N/m}$, $K_{zz} = 100 \text{ N/m}$ for the controller stiffness, $D_{xx} = 20 \text{ Ns/m}$, $D_{zz} = 5 \text{ Ns/m}$ for the controller damping, and $B_{\theta} = 0.025 \text{ kg } \cdot \text{m}^2$ for inertia scaling. The controller stiffness in the $z$-direction is selected much smaller compared to the stiffness in the $x$-direction since $z$-direction is the constrained direction and forces need to be controlled in that direction. The reference trajectories for this experiment are generated using the point-to-point trajectory design method introduced in section 5.2.1. The errors in the end-effector positions and forces for the constrained and unconstrained directions are presented in Figure F.18. It can be observed from this figure that when the manipulator comes to rest neither the position nor the force errors converge to zero. The reasons for this are the static friction in the transmission mechanism and imperfect gravity compensation.

The end-effector forces and positions recorded for the second sliding experiment which are not given in section 6.3.2 are shown here in figures F.19 and F.20 respectively. The straight lines in Figure F.19 are related to the end-effector forces and the dashed lines are the reference trajectory signals. The straight lines in Figure F.20 are related to the end-effector positions calculated from the forward kinematics based on encoder signals, and the dashed lines are the reference trajectory signals.
F.3. Identification of Contact Normal Directions

For tasks such as writing on a blackboard, wiping the table, and similar, the natural position constraint is along the normal direction of contact. The identification of the contact normal in this case can help to automatically select the controller stiffnesses in constrained and unconstrained directions. Identifying the contact stiffness matrix from the recorded position and force data can reveal information about the normal direction of a surface. Singular Value Decomposition (SVD) can be used to calculate the principal directions of the stiffness matrix, thereby to estimate the contact normal. The estimation of the contact normal is needed when the contact normal on the surface does not coincide with any principal direction of the base.
coordinate system of the robot (i.e., when the surface is tilted or the surface is curved). The SVD of a square matrix, $K_{est} \in \mathbb{R}^{N \times N}$, is given as,

$$K_{est} = U \Sigma V^T$$  \hspace{1cm} (F.2)

where the matrix $\Sigma$ represents a diagonal matrix which contains the singular values of $K_{est}$ [39]. The matrices $U$ and $V$ are orthonormal matrices that represent the left and right singular matrices related to $\Sigma$. The matrices $U$, $\Sigma$ and $V$ have the following properties:

1. The length of the columns of $U$ and $V$ are equal to 1 and are perpendicular to each other.
   $$UU^T = U^TU = I, \quad VV^T = V^TV = I.$$

2. The matrix, $\Sigma$ is defined as,
   $$\Sigma = \begin{bmatrix}
   \sigma_1 & 0 & \cdots & 0 \\
   0 & \sigma_2 & \cdots & 0 \\
   \vdots & \vdots & \ddots & \vdots \\
   0 & 0 & \cdots & \sigma_N
   \end{bmatrix}$$

where $\max(\sigma_i) = \sigma_1$, $\min(\sigma_i) = \sigma_N$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N \geq 0$. The parameter $\sigma_i$, represents the singular values of $K_{est}$. In this work, the columns of the matrix $U$ are used in order to determine the constrained and unconstrained directions from the estimated stiffness matrix. An additional sliding over a stiff object experiment has been introduced in this section. The experiment is again related to sliding over a stiff object. The demonstrated task again consists of the same three phases as the sliding experiment introduced in section 6.2. The end-effector forces and positions recorded during the demonstrations are shown in Figure F.21 and F.22, respectively. It can be observed that the force in the $z$-direction exhibits a significant change compared to the forces in other directions while the tip is in contact.

![Figure F.21](image-url)

**Figure F.21** End-effector forces for sliding on a stiff object
The parameters of the virtual soft finger (V.S.F.) that are used during the experiments are $K_v = 700 \, \text{N/m}$ and $B_v = 10 \, \text{Ns/m}$. The parameters of the forgetting factor are $X_H = 0.02 \, \text{m}$ and $T_H = 1 \, \text{sec}$. The uncertainty estimation parameter that is used during the experiment is $\alpha = 0.5$. According to the selection of the forgetting factor parameters, the bound on the low-speed motion becomes $X_H / T_H = 0.02 \, \text{m/s}$. The discontinuities (i.e. gain and loss of contact) are detected using the thresholding on the norm of the force, instead of using the second method mentioned in section 3.3.4. This is because the second method is oversensitive to unmodeled friction effects and detects more discontinuous instants while the tip is in contact with the object. The threshold on the force is selected as 0.5 [N]. The estimation results obtained from the impedance perception algorithm are given in Figures F.23, and F.24. The bias terms that are estimated using the impedance perception algorithm are given in Figure F.23.
The straight lines in these figures represent the estimated components of the impedance determined using the recursive algorithm, the horizontal dash-dot lines are the estimates achieved using the batch least squares algorithm, the gray bands represent the uncertainties of the estimates, and the vertical dotted lines indicate the instants when the discontinuities are detected (i.e. gain and loss of contact in this case). The batch processing is applied when the end-effector is in constrained motion phase. The estimated discontinuity instants are close to the actual moments when the discontinuities occur. This can also be seen from the end-effector force plots in Figure F.21, since after 7.1 second the end-effector forces become different than zero and start increasing, while after 20.3 second the end-effector forces become zero again. The stiffness terms that are obtained using the impedance estimation algorithm are given in Figure F.24.

![Stiffness terms estimated using the recursive/batch algorithms and uncertainties](image)

It can be observed that when the end-effector is in contact with the liftboy the stiffness value $K_{zz}$ is converges close to $K_v$ with a relatively small uncertainty. This is because the surface normal is in the $z$-direction and the object is very stiff. The stiffness in the tangential direction component $K_{xx}$, when the end-effector is in contact with the liftboy, is very close to zero with a relatively small uncertainty. It can also be observed that the off diagonal term $K_{xz}$ converge close to correct zero value, while the other $K_{zx}$ fluctuates with a relatively large uncertainty, the most probably due to kinetic friction acting on the tip while it slides over the object’s surface. Both the bias and the stiffness values in both of the unconstrained motion phases are close to zero with a low uncertainty, excluding the short transients.

SVD is performed only when the end-effector is in contact with the environment. It is calculated at every time instant during the constrained motion phase. In this work, the vectors of the matrix $U$ are used in order to determine the constrained and unconstrained directions from the estimated stiffness matrix. The results of the SVD are presented by using the singular values of the stiffness matrix and the columns of the matrix $U$. In order to determine the direction of the separate columns of $U$, the angle between them and the $x$ - direction of the base frame shown in Figure 6.1 are calculated. These angles are given in Figure F.25.
According to these results the first vector, $u_1$, makes $90^\circ$ with the x-axis which means that it is concurrent with the z-direction of the base frame, whereas the second vector, $u_2$, makes $0^\circ$ with the x-axis, hence it is concurrent with the x-direction of the base frame. It can be observed that second angle shows discontinuities around 11, 13 and 15 seconds. It is caused by the wrapping of the angle during its calculation. However it doesn’t introduce any problems for calculation of the controller stiffness matrix. The calculated singular values of the stiffness matrix, while the end-effector is in contact, are displayed in Figure F.26.

It can be observed from this figure that apart the initial transient effects, the first singular value converges close to $K_v$, whereas the second singular value converges close to zero. This is because the contact stiffness is highest in the contact normal direction whereas it is lowest.
in the tangent directions of the surface. The controller stiffness matrix during the contact phase which takes into account different stiffnesses for constrained/unconstrained directions, is calculated by using

\[ K_x = k_f \, nn^T + k_p \, (I - nn^T) \]  

where \( k_f \) represents the stiffness for constrained directions in which force control is important and \( k_p \) represents the stiffness for unconstrained directions in which position control is important, \( n \) represents the identified contact normal by using SVD. The controller stiffness matrix calculated by using (F.3), where the stiffnesses for constrained/unconstrained directions are selected as \( k_f = 200 \, \text{N/m} \), \( k_p = 2000 \, \text{N/m} \) is displayed in Figure F.27.

![Figure F.27](image)

**Figure F.27** Calculated controller stiffness by using the identified contact normal direction

It can be observed that apart from the initial transient effects the controller stiffness \( K_{xx} \) converges to \( k_p \), since \( x \)-direction is the unconstrained direction, and the controller stiffness \( K_{zz} \) converges to \( k_f \), since \( z \)-direction is the constrained direction. The same reasoning also holds true for the off–diagonal stiffnesses, which are small. The controller setpoints in both directions (i.e. constrained/unconstrained) for replay during the constrained motion phase is calculated by

\[ x_{ref} = x_{demo} + nn^T \, F_{demo} / k_f \]  

where \( x_{demo} \) represents the recorded positions during the demonstrations and \( F_{demo} \) represents the recorded forces during the demonstrations. The calculated setpoints by using (F.4) during the contact phase is given in Figure F.28. The recorded setpoint signals are low-pass filtered and downsampled by a factor of 10 before replay. The calculated controller stiffness matrix is also downsampled 10 times before replay. It can be observed that apart from initial 0.5 seconds, the setpoint in x-direction is very close to the demonstrated one. The setpoint in z-direction is also inside the environment such that the manipulator can apply forces to the environment.
Figure F.28  Calculated setpoints for replay during contact phase