Model-based control of a two degrees-of-freedom haptic gearshift

A. Ayemlong Fokem

CST 2010.025
Abstract

The automotive industry is experiencing not only a strong enhancement of automation in primary driving tasks, e.g., automated gear shifting, lane-keeping systems, adaptive cruise control, park-assists automatic-hold, brake-by-wire, etc, but also, the introduction of more auxiliaries and interior functions, e.g., USB- connectors, mp3 players, navigation systems, among others. These trends lower the driver’s workload significantly, and draw the driver’s attention more and more to interior functions of the car. Adding more functionality to vehicles increases driver satisfaction or pleasure, however, it can also lead to a significant increase of driver distraction. Haptic cues might offer a promising and relatively unexplored alternative to give warnings and other messages to the driver, also to aid drivers in the execution of their driving tasks.

This master project is to research a controlled haptic force feedback shift lever that can accurately reproduce the behavior of a customary gear shift during driving. The haptic interface is a two degrees-of-freedom mechatronic device, whose working principle is based on the self-locking property of a worm pair transmission driven by two electrical motors. Without actuation, it is therefore impossible to force the worm to rotate by applying a force to the lever mounted on the worm gear.

In this research work, we use the Euler-Lagrange formalism and robotica toolbox to obtain the dynamical model of the haptic interface. To incorporate friction in the model, We implement an identification procedure to measure the friction torque, then we select a modified static friction model with Stribeck velocity to approximate the friction model of the haptic device. Thereafter, we obtained the parameters of the friction model, using a nonlinear least square optimization tool from Matlab toolbox. Furthermore, we propose an hierarchical control scheme consisting of two control loops: an inner-loop based on inverse dynamics to control the position of the device, and an outer-loop based on force-feedback using PID control with anti-windup, to enhance the haptic feeling to the operator. Moreover, we implement some simple haptic pattern, an automatic gear shift, etc using Matlab simulink and dSpace software environments.
Acknowledgments

Through my years at Eindhoven University of Technology, many kind, generous and helpful people made my experience one that I will never forget. First, I would like to thank my supervisor Professor Maarten Steinbuch for his guidance, encouragement and support throughout my entire study. Next, I am grateful to my coach Dr. Alexander Serrarens, who gave me the opportunity to carry my Master’s project at DTI Automotive Mechatronics, and moreover for his technical expertise. I am thankful to Dr. Eloisa García Canseco, not only for serving as my coach, but for all of the advice and direction she provided in every phase of my research work.

Special thanks are due to Ir. Corinne Jongeneelen, Dr. Pieter Nuij, Dr. Norddin El Ghouti for asking helpful and insightful questions, and giving valuable advices during my study and research work.

I spent many late nights and early mornings working on assignments with my friends Ir. Alper Denasi and Ir. Michael Ronde with whom I attended some lectures. They provided many words of motivation and encouragement as did other good friends: Ir. Tijs Donkers, Ir. Can Cilli, Ir. Boere Stijn, Ing. Wiebo Kamphuis, Wesley Ooms.

I am thankful to all the staff members of the DCT Lab. They made the lab both academically enriching and a fun place to work.

There are many other friends, family members and loved ones that made the completion of this study phase possible. They are too numerous to mention. Above all, however, I give thanks to God for blessing me with the opportunity to pursue a graduate degree at Eindhoven University of Technology and to see it to completion.
# Contents

Abstract i

Acknowledgments ii

List of Symbols ix

1 Introduction 1

1.1 What is haptic? 2
1.2 Haptic interface: The C-Lever 3
1.3 Haptic interface control 5
1.4 Research objectives 6
1.5 Thesis outline 7

2 Literature survey 9

2.1 Functional structure during gear shifting 9
2.2 The three stages of the gear engagement in manual transmissions 10
2.3 Some research works about haptic gearshift 12

3 Modeling 15

3.1 Kinematic model 15
3.2 The Jacobian 16
3.3 Alternative kinematic representation 17
3.4 Dynamic model of the C-Lever 19
  3.4.1 Rotational inertia 19
  3.4.2 Equations of motion 20
3.5 Discussion 20

4 Friction Identification 21

4.1 Introduction to friction models 21
  4.1.1 Static friction models 21
  4.1.2 Dynamic friction models 23
4.2 Worm-gears transmission 23
4.3 Identification of the friction torque 24
4.4 Friction torque measurements 26
4.5 Friction model approximation 29
CONTENTS

4.5.1 Model selection ........................................ 29
4.5.2 Fitting the data to the friction model ................. 30
4.6 Friction model validation .................................. 31
4.7 Discussion .................................................. 33

5 Position and force-feedback control design .......... 35
  5.1 Control problem formulation ............................... 35
  5.2 Inner-loop controller ..................................... 37
    5.2.1 Joint space inverse dynamics control ............... 37
    5.2.2 Evaluation of the joint space inverse dynamics control ........................................ 39
    5.2.3 Operational space inverse dynamics control .... 43
    5.2.4 Evaluation of the operational space inverse dynamics control ............................... 44
    5.2.5 Operational space control vs joint space control .................................. 49
  5.3 Outer-loop controller ...................................... 50
    5.3.1 Virtual admittance .................................... 50
    5.3.2 Force feedback control using PID with anti-windup ................................. 51
  5.4 Free motion and virtual wall .................................. 51
  5.5 Discussion .................................................. 54

6 Haptic implementation ........................................ 55
  6.1 Haptic patterns ............................................ 55
  6.2 The operator’s force ....................................... 56
  6.3 Experimental results ....................................... 57
  6.4 Discussion .................................................. 57

7 Conclusions and Recommendations .......................... 59
  7.1 Conclusions ................................................ 59
  7.2 Recommendations ........................................... 60

Bibliography ..................................................... 61

Appendices ......................................................... 65

A Specification of the haptic setup ........................... 65

B Model properties ............................................... 67
  B.1 Gravitational forces property used for control gain selection ........................................ 67
  B.2 Inertia matrix elements .................................... 68
  B.3 Coriolis and centrifugal forces matrix elements ........................................ 68
  B.4 Gravitational forces vector elements ........................................ 68
  B.5 Haptic device’s parameters .................................. 69
  B.6 Robotica data file .......................................... 69

C Force sensor’s filter design .................................. 71
CONTENTS

C.1 Force sensors ................................................. 71
C.2 Filtering choice ............................................... 71
C.3 Denoising filter design ....................................... 74
C.4 Filtering results ............................................... 77

D Simulation model overview ........................................ 79

E Matlab files ...................................................... 83
  E.1 Filter parameter’s file ....................................... 83
  E.2 Friction model estimation .................................. 83
List of Symbols

\[ T^0_2 \] Transformation matrix \[-\]
\[ i \] Index, \( i = 1,2 \) \[-\]

**Constants**

\( \delta \) adjusting parameter of the Strubeck curve \[-\]
\( \lambda \) smoothing parameter of the friction at zero velocity \[-\]
\( \mathfrak R \) Kinetic energy \[ J \]
\( \mathfrak L \) Lagrangian \[ J \]
\( \mathfrak U \) Potential energy \[ J \]

\( \omega_{ni} \) natural frequencies \[ \text{rad/s} \]
\( \tau \) Torque in generalized coordinates \[ \text{Nm} \]
\( \tau_c \) Coulomb friction \[ \text{Nm} \]
\( \tau_f \) friction torque in generalized coordinates \[ \text{Nm} \]
\( \tau_s \) static friction \[ \text{Nm} \]
\( \tilde f \) force error \[ \text{N} \]
\( \vec r_i \) radius vector of center of Mass \( [r_{ix}, r_{iy}, r_{iy}]^T \) \[ \text{m} \]
\( \zeta_i \) damping ratio \[-\]

\( C(q, \dot q) \) Coriolis and Centrifugal matrix \[ \text{Nms/rad} \]
\( f_d \) desired force \[ \text{N} \]
\( F_e \) frequency of the virtual dynamics \[ \text{Hz} \]
\( f_e \) virtual environment force \[ \text{N} \]
\( f_h \) human hand force \[ \text{N} \]
\( f_{lim} \) maximum force \[ \text{N} \]
\( f_{ni} \) bandwidth control frequencies \[ \text{Hz} \]
\( g \) Gravity acceleration \[ \text{m/s}^2 \]
\( g(q) \) Gravitational torque vector \[ \text{Nm/rad} \]
\( J_i^\dagger(q) \) pseudo-inverse of the linear velocities Jacobian matrix \[-\]
\( J_1, J_2 \) Moment of inertia of link 1 and 2 [Nms^2/rad]

\( J_w(q) \) Jacobian matrix of the angular velocities [Nms/rad]

\( J_v(q) \) Jacobian matrix of the linear velocities [Nms/m]

\( K_D \) derivative gain matrix [Nms/rad]

\( K_{fa} \) acceleration feedforward gain matrix [Nms^2/rad]

\( K_f \) force gain matrix [N/m]

\( k_g \) gravitational torque constant [Nm/rad]

\( K_P \) proportional gain matrix [Nm/rad]

\( K_w \) virtual wall gain [N/m]

\( M(q) \) Moment of inertia matrix [Nms^2/rad]

\( m_i \) Mass of link \( i \) [kg]

\( T_d \) derivative action time constant of the PID [-]

\( T_i \) integral time constant of the PID [-]

\( T_t \) tracking time constant of the PID [-]

\( T_{wd} \) derivative action time constant of the virtual wall [-]

\( X_p \) set of kinematic coordinates [m]

\( b \) Viscous friction coefficient [Nm/s]

**Variables**

\((x_i, y_i, z_i)\) coordinates frame [m]

\( \alpha_i \) link twist [deg]

\( a_i \) link length [m]

\( d_i \) link offset [m]

\( q \) Generalized coordinate \([q_1, q_2]^T\) [rad]

**Abbreviations**

\( CM_i \) Center of mass \( i \)

AMT Automated Manual Transmission

C-Lever Clever Lever

DOF Degree of freedom
DTI-AM DTI Mechatronics

EL  Euler – Lagrange

HUD  head-up display

VE  virtual environment
Chapter 1

Introduction

In the automotive industry, safety, comfort and environmental issues strongly influence the development of automobiles. Research capacity and money is invested in new technologies to keep track of these issues. Transmission technology, for example, plays an important role in reducing the energy consumption. Automated gear shifting, lane-keeping systems, adaptive cruise control, park-assists, automatic hill-hold, brake-by-wire, and useful warning features lower the driver’s workload significantly and increase the driver’s comfort. Therefore the driver’s attention is drawn more and more to the interior functions of the car. Adding more functionality to vehicles increases driver satisfaction or pleasure, however, it can also lead to a significant increase of driver distraction. Haptic cues might offer a promising and relatively unexplored alternative to give warnings and other messages to the driver, and also to aid drivers in the execution of their driving tasks. Haptic interface is an emerging technology that has been accepted by the automotive industry as a user interface in cars. Haptic interface provides an independent sensory channel (which brain can process relatively faster than vision) and enhances user’s experience in a multimodal environment. Several studies have shown that the human ability to discriminate using the sense of touch does not impair contrary to sight or hearing with increasing age (Xia et al., 2002). The best example lies in the aircraft technology, where the mechanical controls were replaced by electronic controls in 1980’s. A haptic system was integrated to provide feedback for the gravitational forces experienced by flaps and tail during flight. In the automotive industry, various types of input devices have already been placed on steering wheels. A very simple example is the cruise control on the steering wheel, for which driver does not have to take his hands off yet he can control the acceleration/ deceleration of the vehicle. But, current gear selector levers for transmission gearboxes in passenger cars are mostly large mechanic constructions connected to the gearbox by a damped Bowden cable\(^1\). These traditional components are large, heavy, slow, uncomfortable to use, and difficult to integrate in center consoles. Because of these disadvantages, electronic gear selector levers were created. Some applications positioned behind the steering wheel (Sekiguchi et al.) or on the dash board (Burkhardt et al.), such as with push buttons, can be found on the market. The development of the gearshift lever, could use some experience with the shifters for automated manual transmission (AMT) that are based on the mechanic levers, but without a Bowden cable (shift-by-wire) (Buchs et al.). Shift-by-wire is about replacing the mechanical link between the gearbox and the shift lever with an electromechanical system or mechatronics device. This mechatronics device will make new safety functions possible and will assist the driver during gear shifting, while providing the same feeling as the normal manual gear shifting. Furthermore, shifting patterns can be varied\(^2\) features.

---

\(^1\)A Bowden cable is a type of flexible cable used to transmit mechanical force or energy by the movement of an inner cable (most commonly of steel or stainless steel.) relative to a hollow outer cable housing.

\(^2\)according to the user preference
1.1 What is haptic?

The term "haptic" comes from the Greek word *hapteshtai*, which refers to the sense of touch.

The ability to recognize objects by touch can be categorized into *tactile* and *kinesthetic* cues received throughout the body.

**Tactile** Relating to touch or to the sense of touch (Mifflin, 1995). The sense by which pressure or traction exerted on the skin is recognized, the sense by which the properties of bodies are determined by contact. For example when feeling the roughness of a surface.

**Kinesthetic** The sense perception of movement, the muscular sense (Mifflin, 1995). Often used to describe the sensory system in the human body that records limb positions, forces and joint torques. Examples of kinesthetic cues would be contours, shapes, and sensations like the weight of the glass filled with water, the resistance of the piano keys, or the impact of hitting a tennis racquet’s sweet spot.

Tactile and kinesthetic cues are received from *mechanoreceptors* present throughout the human body. Mechanoreceptors are specialized sensory end organs that respond to mechanical stimuli such as tension or pressure (Mifflin, 1995). If mechanoreceptors would stop functioning, everyday tasks like feeling a rough surface may suddenly become impossible. In some cases, for example when your leg has fallen asleep or when tying your shoe laces after playing in the snow, the mechanoreceptors do not function properly. The difficulty and clumsiness which occurs in these examples, arise due to mechanoreceptors that do no longer sending critical information as touch, pressure, stretching and motion to the brain.

In Engineering, *haptic technology* refers to the means or knowledge to develop, design and build systems that are able to provide/present information to the user by the sense of touch. The general process of communicating information back to the user using force signals is referred to as *haptic feedback*. Feedback in the sense of the human-human or machine-human communication and/or interaction by means of force. Figure 1.1 presents a schematic view of such a machine-human interaction adopted for the gearshift lever. The input of the haptic interface consists of the operator acting on the end effector (lever knob), which is monitored by the sensing system. After evaluation of this information by the computer (control system and virtual environment), the output of the haptic interface is a reaction to the operator through the actuation system. The end effector input and output can together be regarded as interaction. The interaction between operator and the virtual environment through the haptic interface is a combination of force and displacement. Since force times displacement represents mechanical work, the interaction can be seen as a bidirectional transfer of energy. Energy transfer, if not properly controlled, may have a detrimental, destabilizing effect on the simulation (Burdea et al., 2003). Thus, a very important issue related to haptic interfaces is the control system.

![Figure 1.1: Human interaction with a haptic interface.](image)
1.2 Haptic interface: The C-Lever

A haptic interface is a device configured to provide haptic information to the operator. Just as a video interface allows the user to see a computer generated scene, a haptic interface permits the user to "feel" the virtual environment. Haptic displays generate forces and motions, which are sensed through both touch and kinesthesia (Adams and Hannaford, 2002).

![C-Lever concept](image1)

![2-DOF experimental device](image2)

Figure 1.2: Two degrees-of-freedom haptic gearshift device

The haptic interface, as depicted in figure 1.2 is a two degrees-of-freedom mechanical device, driven by two electrical motors. Different from tele-operation systems, the C-Lever is one device combining both master and slave: the user operates the lever and at the same time this lever moves and returns a haptic force that is influencing the movement again. Its working principle is based on the self-locking property of a worm pair transmission (Serrarrens, 2007). Without actuation, it is impossible to force the worm to rotate by applying a force to the lever. That is, the operator experiences infinite stiffness of hitting a wall when there is no current fed to the actuators (electrical motors). Vice versa, the actuators are able to move the worm gear and lever around their rotation axes rather easily. The transmission ratio of the worm pair is very large, offering a reduced workload on the actuators and a large force feedback on the lever knob (end effector). The main advantage of this worm pair transmission is the low power consumption with a large force feedback in a constrained workspace. On each motor output axis, an encoder is mounted to measure rotational velocity and position. These motors are permanent magnet motor, driven by two H-bridge converters. The converters are series of miniature servo amplifiers for DC brush current, that incorporate custom mixed analog/digital ICs and a hybrid power stage. The size of the workspace of the C-Lever is $\pm 4$ cm in the X-axis and $\pm 4$ cm in the Y-axis, and the actuators strokes are 15 degrees for both joints.

The C-Lever is an admittance controlled device, it measures the force exerted by the human hand on the lever knob using four force sensors fixed orthogonally into the force sensor base attached on the lever stick. The force sensors are piezo-electric sensors, called FlexiForce sensors. The sensors are ultra-thin and flexible printed circuit, which can be easily integrated into most applications. With their paper-thin construction, flexibility and force measurement ability, these sensors can measure force between almost any two surfaces and are durable enough to stand up to most environments. The scale of the output voltage of the sensors is calibrated to represent the operator’s force range of 0 to 4 N, and the induced noises are eliminated by some filtering scheme build on the hardware or in the software environment. The link between the hardware and the software is provided by the hardware server and the controller board that come along.

---

3 An H-bridge is an electronic circuit which enables a voltage to be applied across a load in either direction. These circuits are often used in robotics and other applications to allow DC motors to run forwards and backwards.
with the $dSpace^\text{©}$ software. The DS1102 controller board provides the input and output ports. This controller board is connected via a 62-pin SUB-D connector cable to the hardware server, which is built as a standard PC card. The $dSpace^\text{©}$ software contains the Controldesk testing environment, the RTLib1102 real-time interface Simulink$^\text{©}$ library blockset and the Texas Instrument TMS320C ANSI compiler. Figure 1.3–1.6 illustrate the body parts of the C-Lever.
1.3 Haptic interface control

Haptic interfaces have two basic functions (Salisbury and Srinivasan, 1992; Burdea and Grigore, 1996). The first is to measure forces, positions, and their time-derivatives at the operator’s hand (or other body locations). The second function is to display forces and positions back to the operator. There exist two fundamental methods for controlling haptic interfaces (Burdea and Grigore, 1996). When position is input to the control loop and forces are fed back to the operator, we speak of force-feedback control. Alternatively, the simulation can use position-feedback control, in which forces applied by the operator are sensed and positions are fed back through the haptic interface. According to these two different control methods, haptic interfaces are categorized into impedance controlled and admittance controlled devices. Figure 1.7 shows both fundamental control methods.

![Figure 1.7: Fundamental methods for controlling haptic interfaces](image)

Impedance controlled devices use force-feedback control (figure 1.7(a)), so the operator moves the haptic interface, and the device will react with a force if a virtual object is met (Linde et al., 2002). The operator will inevitably feel the mass and the friction of the actual device, but these can be made very small by careful mechanical design. Impedance devices are by nature lightly built and highly back drivable (meaning that the operator can move the end effector effortlessly when no virtual objects are met). They are typically cable driven by high performance DC motors.

Admittance controlled devices use position-feedback control (figure 1.7(b)), so the operator exerts a force on the haptic interface, and the device will react with the proper displacement. Admittance control allows considerable freedom in the mechanical design of the interface, because backlash and mass can be practically eliminated (Lammertse et al., 2002). As a result, the mechanism can be quite robust, and capable of displaying high stiffness and high forces. Impedance control and admittance control are dual not only in their cause-and-effect structure, but also in their performance. The impedance controlled device is typically lightweight, backlash free, and renders low mass (Adams and Hannaford, 2002; Linde et al., 2002). Consequently, performance is lacking in the region of higher forces, high mass, and high stiffness. Adding complex end effectors is also a problem. Admittance controlled interfaces on the other hand, are capable of rendering very high stiffness and minimal friction. They are very suitable for larger workspace and for carrying complex end effectors with many degrees of freedom. Moreover, because forces are sensed rather than computed in real time, admittance control has the advantage of reduced modeling computation load (Burdea and Grigore, 1996). Compared to impedance control, a given computing platform will therefore have an increase in the high-level feedback loop bandwidth when admittance control is implemented. However, admittance controlled interfaces are often not capable of rendering very low mass. The minimal mass rendered at the end effector (the minimal mass the operator feels during the simulation) is called the minimal tip inertia.
The haptic interface used to develop this project, is an admittance controlled device. The reason for this choice is that in order to mimic the gearshift feel of a manual operated transmission, it is expected that high forces and high stiffness need to be rendered, which an impedance controlled device is not capable of. Moreover, during gear shifting, the operator performs quick movements, which result in great accelerations of the gear knob. Due to its increased high-level feedback loop bandwidth, it is expected that an admittance controlled interface offers a more accurate and more stable simulation.

1.4 Research objectives

This research work is part of the haptic gearshift project, supported by the SenterNovem IOP ManMachine Interaction Project MMI07105 ”Control solutions for humanintheLOOP user interfaces” and by Drivetrain Innovations B.V. The goal of the haptic gearshift project is to research a controlled haptic force feedback shift lever that can accurately reproduce the behavior of a customary gear shift during driving. The objectives of the research work are:

1. Modeling and friction identification of the haptic interface

   The worm pair transmissions are necessary to reduce the workload of the actuators, but contain several nonlinearities such as, stiction, Striebeck effects, Coulomb friction, backlashes and torque ripples. Stiction can cause limit cycling. Coulomb friction, which can extend system stability by absorbing the oscillation energy, may lead to input-dependent instability and actuator limit cycles. Friction not only limit performance in precise positioning, but also make high performance force control more difficult. Nevertheless, the consequences of friction in mechanical transmission can be minimized either by a mechanical system design, or masked by control design (Dohring et al., 1993). Standard methods to obtain mathematical model of manipulators are diversely explained in books of robot manipulators (Dombre and Khalil, 2007; Kelly et al., 2001; Spong and Vidyasagar, 1989), and so are methods and studies about friction and friction model, mostly in some literature (Olsson et al., 1998; Lampaert et al., 2002; Swevers et al., 2000; Barahanov and Ortega, 2000). We utilize some of these methods to elaborate the complete dynamical model of the C-Lever.

2. Position and force-feedback control design

   Using the complete dynamical model of the C-Lever, we design a controller that computes an input current/torque to the motors, to eliminate the effects of self-locking properties and other undesired dynamics effectively. The elimination of these undesired dynamics allows a smooth free-motion of the worm-pair transmission when a shifting effort is applied by the operator. Besides, the design of a force feedback control that eliminate the operator’s hand disturbances and enhance the haptic feeling by correcting the interaction forces at the lever knob, is also investigated.

3. Haptic patterns implementation (virtual environment)

   The implementation and testing of an automatic gearshift pattern is further investigated to conclude the research work of the project by experimental evaluation.
1.5 Thesis outline

Chapter 2 presents the survey of some literature works in haptic, and particularly in haptic for automotive applications. The kinematics and the dynamics model of the C-Lever follow in Chapter 3. A discussion about the method and the achieved models concludes the chapter. Chapter 4 deals with the experimental identification of friction. The chapter opens with an introduction to friction models. Next, a control law is designed to measure the friction torque. The chapter further discusses the approximation of the friction model, followed by the friction model validation. A discussion about the motivation of the friction identification, the method and results concludes the chapter. Two control loops are then designed and evaluated in Chapter 5 for position control and for force-feedback control. Moreover, a free motion and virtual wall are implemented to complete the evaluation of the overall control design. Some comments are then presented in the discussion section to conclude the chapter. Chapter 6 presents the implementation results of some haptic patterns (e.g., automatic gearshift, etc), followed with some comments on the practical limitations of this implementation. The conclusion about the research objectives and the achieved results is presented in chapter 7. Some open questions are furthermore presented in the recommendations section of this chapter.
Chapter 2

Literature survey

This chapter presents a review of some literature about haptic interfaces in automotive applications.

2.1 Functional structure during gear shifting

(Tideman et al., 2004) drew up a functional structure gearbox operation based on the analysis of gearbox construction and operation. The operator exerts a certain force on the gear knob. When this force is larger than the resistance of the gearshift system, the gear knob is moved. At the same time, the operator receives feedback through receptors in the skin, muscles, tendons and skeletal joints (tactile sense and proprioceptive system). The visual sense and auditory sense also provide feedback on the progress of the shifting action. Figure 2.1 illustrates this interaction.

![Diagram: Functional structure of a manual transmission in a passenger car](Adapted from (Tideman et al., 2004))
Converting force and motion While holding the gear knob, the operator is able to perform all kinds of movements within a certain workspace. Through the gear lever and a ball joint, these movements are converted into two practically independent movements: the movement of the selector cable and the movement of the shifting cable.

Transferring force and motion The selector cable and the shifting cable transfer both movements to the gearbox.

Selecting a shifting hub The movement of the selector cable results in selection of a particular shifting hub.

Selecting and engaging a gear ratio After a shifting hub is selected by the movement of the selector cable, the movement of the shifting cable results in selection and engagement of a particular gear ratio. Yet, it shows that there is a certain dependency between both movements: engagement of a gear ratio is only possible after the corresponding shifting hub is selected.

2.2 The three stages of the gear engagement in manual transmissions

(Frisoli et al., 2001a,b) studied and presented on figure 2.2 and 2.3 the three different main stages, occurring during the gear engagement. These three stages characterize the particular force response of a gear-shift: the synchronization, the engagement and the impact against the mechanical stop. During the synchronizing phase both the force and the position are held constant, as shown in figure 2.2. The force reaches its maximum value, and the position is held constant for a definite period of time. The engagement stage is characterized by an isolated peak force, that is lower than the synchronizing force peak. Moreover the magnitude of such peak force is variable, so that engagement peaks can vary remarkably. As shown in figure 2.3 the synchronizing and the engagement peaks occur at definite values of the x position. In particular the synchronizing stage reaches a peak value of about 8 kg, and has a duration of about 0.3 ms. The engagement peak is more instantaneous, since it is due to the impact of the meshing. The forces that the driver exerts on the lever, when he changes into a gear, are so determined mainly by the stage of synchronizing, engagement and stop impact.
2.2. THE THREE STAGES OF THE GEAR ENGAGEMENT IN MANUAL TRANSMISSIONS

Figure 2.2: Force and position vs. time, as measured at the knob during neutral-first gear shift
(Adapted from (Frisoli et al., 2001a))

Figure 2.3: Force vs. position, as measured at the knob during neutral-first gear shift
(Adapted from (Frisoli et al., 2001a))
2.3 Some research works about haptic gearshift

(Frisoli et al., 2001a,b) employed in their research work a two degrees-of-freedom force-feedback joystick to simulate the force response of a manual gearshift of car during driving. The control law was based on an hybrid model. A state machine determines the active state of the system, according to the simulation, and changes the parameters of the dynamical model. An analytical model of the gearshift behavior was synthesized to replicate a correct force-feedback to the operator. The different phases of the gear shift modeled through a dynamic model with both continuous and discrete states (each discrete state was associated to a gearshift stage). The gearshift response was then developed as a MATLAB Simulink®/Stateflow module, to represent an hybrid model.

(Elixabete and S., 2008) developed in their research work, a cost effective haptic device that can be used for cars, trains, trucks, etc, with a capacity to render forces equivalent to the forces exhibited by real controls. Two haptic devices that use force control to accurately reproduce the behaviour of a lever and gear shift for driving simulators were investigated. The first device is a one rotational degree-of-freedom lever to emulate the behaviour of train and tram controls (Figure 2.4). The second one is a two degrees-of-freedom haptic device specially designed to simulate different gearshift behaviours (Figure 2.5). DC brushed motors with encoders transmit torque to linkages through pretensioned cable reductions. The use of cable transmissions allows for high mechanical performance. These kinds of mechanical transmissions present high reversibility, high stiffness, no backlash and near zero friction, allowing the system to reproduce realistic haptic sensations. The maximum rotation angle of the 1-DOF haptic lever is ±70 degrees. The size of the workspace of the 2-DOF haptic device is similar to the one of a real manual gearshift with a workspace of ±60 mm in the X-axis and ±40 mm in the Y-axis. In this case, the actuator strokes are 41 degrees for the first DOF and 55 degrees for the second DOF.

Figure 2.4: One degree of freedom train/tram haptic lever
(Adapted from (Elixabete and S., 2008))
2.3. SOME RESEARCH WORKS ABOUT HAPTIC GEARSHIFT

(Lindner and Tille, 2010) designed in their research work a highly integrated mechatronic gear selector lever, illustrated in figure 2.6, for the center console of the BMW X5 / X6 series. The mechatronic lever consists of two general parts: the switching box and the lever knob. The switching box contains the functional elements, and the lever knob contains design and interface elements. Actuators for the haptic user guidance are integrated in the switching box. The different driving modes are symbolized by the first letter of the shift pattern: P (park), D (drive), N (neutral), R (reverse), S (sport), and M (manual). For the position P, a separate push button is integrated into the upper surface of the lever. Because the position P is not in the shift pattern anymore, the position R is the last one in the automatic rail. As a result, maneuvering the car into a parking space is much more comfortable and faster. For switching into reverse, the lever has to be moved up against the mechanical block only. Both levers are monostable which means that after release, the lever returns into a mechanical zero position. The first advantage of being monostable is that functions (such as automatic P after engine stop) can be automated, independently of the lever’s current position, since the lever always returns to the same zero position. The second advantage of monostability is that the lever can always be found at the same position intuitively while driving.
(a) Mechatronic gear selector for BMW X5 / X6 series.

(b) 3-D view of the assembly of the mechatronic lever box of the BMW X5 / X6 series.

Figure 2.6: Mechatronic gear lever for an AT gearbox designed for implementation in the center console of the BMW X5 / X6 series (Adapted from (Lindner and Tille, 2010)).
Chapter 3

Modeling

In this chapter we derive the kinematics and the dynamic model of the C-Lever, using the forward kinematics problem and the Euler-Lagrange framework, respectively.

3.1 Kinematic model

The problem of forward kinematics consists in describing the motion of the manipulator without consideration of the forces and torques causing the motion. This forward kinematics is utilized to determine the position and orientation of the end-effector (lever knob) given the values for the joint variables of the C-Lever.

We develop a set of conventions that provide a systematic procedure for performing this analysis using the well-known Denavit-Hartenberg convention or D-H parameters. It is of course, possible to carry out forward kinematics analysis even without respecting these conventions, as previously done in (Weel, 2007; van Diepen, 2008); however using the D-H convention simplifies the analysis considerably. The D-H convention is a commonly used convention for selecting frames of reference in robotic application (Spong and Vidyasagar, 1989). In this convention, each homogeneous transformation is represented as a product of four basic transformations.

Figure 3.1: D-H representation: Coordinate frame attached to the C-Lever
We number the joints from 1 to 2 and the links from 0 to 2, starting from the base, as shown in Figure 3.1. By this convention, joint 1 connects link 0 to link 1, and joint 2 connects link 1 to link 2. We consider the location of joint $i$ to be fixed with respect to link $i-1$. When joint 1 or joint 2 is actuated, link 1 or link 2 moves, respectively. Therefore, link 0 (the base frame) is fixed and does not move when joints are actuated. We also rigidly attach a coordinate frame to each link. In particular, we attach $0_i x_i y_i z_i$ to link $i$ as required in the D-H convention, and shown in Figure 3.1. This means that, whatever motion the lever executes, the coordinates of each point on link $i$ are constant when expressed in the $i$th coordinate frame. Furthermore, when joint $i$ is actuated, link $i$ and its attached frame, $0_i x_i y_i z_i$, experience a resulting motion.

The frame $0_0 x_0 y_0 z_0$, which is attached to the CLever base, is referred to as the inertial frame.

**D-H parameters**

Based on the coordinate frames configuration from figure 3.1, the corresponding D-H parameters associated with link $i$ and joint $i$ are obtained in Table 3.1. For the four quantities, $q_i$ denotes the joint angle, $a_i$ denotes the length or the distance between the axes $z_{i-1}$ and $z_i$; $d_i$ denotes the offset or the distance between the origin $O_0$ and the intersection of the $x_1$ axis with $z_0$, and $\alpha_i$ denotes the twist determine from $z_{i-1}$ and $z_i$ by the right-hand rule.

<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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>90°</td>
<td>0</td>
<td>$q_1$</td>
</tr>
<tr>
<td>2</td>
<td>$\ell$</td>
<td>90°</td>
<td>0</td>
<td>$q_2$</td>
</tr>
</tbody>
</table>

The corresponding transformation $T_2^0$ matrix is,

$$
T_2^0 = \begin{bmatrix}
\cos(q_1) \cos(q_2) & \sin(q_1) & \cos(q_1) \sin(q_2) & \ell \cos(q_1) \cos(q_2) \\
\cos(q_2) \sin(q_1) & -\cos(q_1) & \sin(q_1) \sin(q_2) & \ell \cos(q_2) \sin(q_1) \\
\sin(q_2) & 0 & -\cos(q_2) & \ell \sin(q_2) \\
0 & 0 & 0 & 1
\end{bmatrix},
$$

where $T_2^0$ is the transformation matrix of any point of the lever’s knob with respect to the base frame of the C-Lever. The first three rows of the fourth column in equation of the transformation matrix represent the forward kinematics of the C-Lever, which is denoted as:

$$
X_p = \begin{bmatrix}
\ell \cos(q_1) \cos(q_2) \\
\ell \cos(q_2) \sin(q_1) \\
\ell \sin(q_2)
\end{bmatrix}.
$$

From these kinematic’s relations, the workspace of the C-Lever is computed and presented in figure 3.3.

**3.2 The Jacobian**

Mathematically, the forward kinematic equations define a function between the space of cartesian positions and orientations and the space of joint positions. The velocity relationships are then determined by the Jacobian of this function. The Jacobian is a matrix that generalizes the notion of the ordinary derivative of a scalar function. The Jacobian is one of the most important

---

4We have used Robotica© to compute Kinematics, Jacobian, and the equations of motion. Robotica© is a program developed by (Netthey and Spong, 1994) to compute the kinematics and the dynamic equations of motion for robot manipulators in mathematica.
quantities in the analysis and control of robot motion. It rises in virtually every aspect of robotic manipulation: in the planning and execution of smooth trajectory, in the determination of singular configurations, in the execution of coordinated motion, in the derivation of the dynamic equations of motion, and in the transformation of forces and torques from the end effector to the manipulator joints (Spong and Vidyasagar, 1989). The Jacobian for the set of kinematics is given (3.1).

\[
J(q) = \begin{bmatrix}
J_v \\
J_\omega
\end{bmatrix} = \begin{bmatrix}
-\ell \cos(q_2) \sin(q_1) & -\ell \cos(q_1) \sin(q_2) \\
\ell \cos(q_1) \cos(q_2) & -\ell \sin(q_1) \sin(q_2) \\
0 & \ell \cos(q_2) \\
0 & \sin(q_1) \\
0 & -\cos(q_1) \\
1 & 0
\end{bmatrix},
\]

where \( J_v \) and \( J_\omega \) are the sub-jacobian matrices describing the linear and the angular velocities of the end-effector (knob) respectively, \( K(q) \) denoting the 6 \( \times \) 2 forward kinematics.

**Kinematics singularity**

Let consider \( J_v \), the sub-jacobian matrix that describes the linear velocity in the workspace, and whose inverse will be used in the control law in Chapter 5. The inverse of \( J_v \) exists if and only if its determinant is not zero (Spong and Vidyasagar, 1989; Vukobratovic et al., 2003; Siciliano et al., 2009). Otherwise \( J_v \) is said to be singular and not invertible. At the singular configuration, \( J_v \) is also rank deficient. The set of coordinates for which \( J_v \) is singular for the C-Lever’s workspace is \((q_1, q_2) = (0, \frac{\pi}{2})\), where

\[
\text{rank}(J_v) = \text{rank} \begin{bmatrix} 
-\ell \cos\left(\frac{\pi}{2}\right) \sin(0) & -\ell \cos(0) \sin\left(\frac{\pi}{2}\right) \\
\ell \cos(0) \cos\left(\frac{\pi}{2}\right) & -\ell \sin(0) \sin\left(\frac{\pi}{2}\right) \\
0 & \ell \cos\left(\frac{\pi}{2}\right)
\end{bmatrix} = 1.
\]

\( \text{rank}(J_v) < \text{dim}(J_v) \), with \( \text{dim}(J_v) = 2 \) at \((q_1, q_2) = (0, \frac{\pi}{2})\).

In the workspace, this singular configuration correspond to the upright position, which is also the homing position for the C-Lever and consequently, the rest position for the controller. Carrying the obtained kinematics further in the design of the feedback control law will result in instable trajectories.

### 3.3 Alternative kinematic representation

The kinematics obtained using the D-H convention have shown some existing singularity in the upright position of the lever, which is also the rest position for the controller. We therefore define a new set of kinematics, using the rotation matrix to eliminate that singularity from the upright position. In the rotation matrix, the coordinates are defined using directly the rotation axes \( q_1 \) and \( q_2 \). Then, the forward kinematic is derived as followed from the configuration in figure 3.2,

\[
X_p = \begin{bmatrix}
\ell \cos(q_1) \sin(q_2) \\
-\ell \sin(q_1) \\
\ell \cos(q_1) \cos(q_2)
\end{bmatrix}
\]

The linear velocity matrix is also derived as,

\[
J_v(q) = \begin{bmatrix}
-\ell \sin(q_2) \sin(q_1) & \ell \cos(q_1) \cos(q_2) \\
-\ell \sin(q_1) \\
-\ell \sin(q_1) \cos(q_2) & -\ell \cos(q_1) \sin(q_2)
\end{bmatrix}.
\]
\[ \text{Rank}(J_v) = 2 \] for \((q_1, q_2) = \left(0, \frac{\pi}{2}\right)\), and corresponds to the dimension of the system \((\text{dim} = 2)\). Therefore, this set of kinematics does not suffer from singularity in the upright position of the lever.

![Image](image1.png)

Figure 3.2: Coordinates frame using rotation matrix

**C-Lever’s workspace**

Both kinematic models describe a sphere, and the workspace of the C-Lever is located at the top of that sphere as shown in light gray area on Figure 3.3. The movement of the lever is therefore constrained in x axis by ±4 cm and in y axis by ±4 cm, while no rotation or translation is possible in z axis.

![Image](image2.png)

Figure 3.3: C-Lever’s workspace
3.4 Dynamic model of the C-Lever

The dynamic model or equations of motion, describe the relationship between force/torque and motion. The equations of motion are important to consider in the design of the control algorithm. The moving parts of the C-Lever can be represented by two rigid bodies, the lever (figure 3.5) and the rotating frame (figure 3.4), both connected by two perpendicular joints, $q_1$ and $q_2$.

![Figure 3.4: body 1: rotating frame](image)

![Figure 3.5: Body 2: Lever](image)

### 3.4.1 Rotational inertia

The rotating frame and the lever are not symmetric bodies, but we consider the inertia of the principal axes only to simplify the analysis. The corresponding moment of inertia are called the principal moments of inertia, since the off-diagonal terms of the inertia tensor matrix are neglected. Figure 3.4 illustrates the center of mass of the rotating frame. Its principal moments of inertia are presented as followed,

$$J_1 = \begin{bmatrix} J_{1x} & 0 & 0 \\ 0 & J_{1y} & 0 \\ 0 & 0 & J_{1z} \end{bmatrix},$$

where $J_{1x}$, $J_{1y}$, $J_{1z}$ are the moment of inertia of the rotating frame with respect to $x$, $y$, and $z$ axes in the workspace respectively. The vector radius of the center of mass with respect to these axes for the rotating frame is also given as followed,

$$r_1 = \begin{bmatrix} r_{1x} \\ r_{1y} \\ r_{1z} \end{bmatrix}.$$

Figure 3.5 illustrates the center of mass of the lever. Its principal moments of inertia are presented as followed,

$$J_2 = \begin{bmatrix} J_{2x} & 0 & 0 \\ 0 & J_{2y} & 0 \\ 0 & 0 & J_{2z} \end{bmatrix},$$

where $J_{2x}$, $J_{2y}$, $J_{2z}$ are the moments of inertia of the lever with respect to $x$, $y$, and $z$ axes in the workspace respectively. The vector radius of the center of mass with respect to these axes is also given as followed,

$$r_2 = \begin{bmatrix} r_{2x} \\ r_{2y} \\ r_{2z} \end{bmatrix}.$$
3.4.2 Equations of motion

For each joint of the C-Lever, the kinetic energy is given by
\[K(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q}\]
with \(M(q) \in \mathbb{R}^{2 \times 2}\) the symmetric, positive-definite inertia matrix, and the potential energy is denoted by \(U(q)\). Hence, applying the Euler-Lagrange formalism (Spong and Vidyasagar, 1989; Siciliano et al., 2009; Dombre and Khalil, 2007; Kelly et al., 2001), the dynamic model of the C-Lever is obtained in a compact notation form as
\[M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \tau_f = \tau,\] (3.4)
where \((q, \dot{q}) \in \mathbb{R}^2 \times \mathbb{R}^2\) represents the system’s state (consisting of generalized joint positions and velocities), \(g(q) = \frac{\partial}{\partial q} U \in \mathbb{R}^2\) denotes the gravitational forces, \(C(q, \dot{q})\dot{q} \in \mathbb{R}^2\) represents the Coriolis and centrifugal forces, \(\tau_f\) denotes the 2 × 1 vector of friction torques, and \(\tau\) denotes the 2 × 1 vector of input torques.

The expressions for inertia matrix \(M(q)\), the coriolis and centrifugal forces \(C(q, \dot{q})\dot{q}\), and the gravitational force \(g(q)\) are derived using Robotica©, and presented in Appendix B.1. The parameters of the C-Lever are computed using a computer-aid program, Pro−Engineer© (Churches and Magin, 1994; Wang, 2009) and also presented in Appendix B.1. Pro−Engineer© is a parametric, integrated 3D CAD/CAM/CAE solution created by Parametric Technology Corporation (PTC).

3.5 Discussion

The kinematics and the dynamic model of the C-Lever were studied in this chapter, using the D-H conventions and Euler-Lagrange formalism, respectively. The analysis of the derived kinematics based on D-H convention, reveals a singularity in the upright position of the lever. Nonetheless, another set of kinematics was derived using rotation matrix, to eliminate the singularity in the control law. Furthermore, the elements and the parameters of the dynamic model were derived using Robotica©, and Pro−Engineer©, respectively.
Chapter 4

Friction Identification

Friction is an important aspect of many control systems both for high quality servo mechanisms and simple pneumatic and hydraulic systems. Friction can lead to tracking errors, limit cycles, and undesired stick-slip motion. Control strategies that attempt to compensate for the effects of friction, without resorting to high gain control loops, inherently require a suitable friction to predict and to compensate for the friction. These types of schemes are therefore named model-based friction compensation techniques. (Canudas de Wit et al., 1995). A good friction model is also necessary to analyze stability, predict limit cycles, find controller gains, perform simulations, etc. This chapter gives a brief introduction about friction models, followed by the identification of the friction model of the worm pair’s transmission of the C-Lever.

4.1 Introduction to friction models

Friction appears at the physical interface between two surfaces in contact. Lubricants such as grease or oil are often used but there may also be a dry contact between the surfaces (Olsson et al., 1998). The technical challenges associated with modeling and friction stem from a couple of sources. First, there is the problem that dry friction is highly nonlinear, posing problems for accurate modeling. High bandwidth and servo stiffness are required to accurately reproduce the transition from a rapid buildup of force just prior to sliding to the break-away as sliding commences. Moreover, the largest nonlinearity, i.e., the discontinuity between static and dynamic friction, occurs at essentially zero velocity, when the signal to noise ratio in the sensed velocity is worst. To overcome these difficulties, research studies have investigated and classified friction in some friction models. A brief introduction on those friction models is presented in this section.

4.1.1 Static friction models

Static friction is a force which resists the lateral movement of two objects which are touching each other. A simple example of static friction might be a wooden block sitting on a ramp. Unless sufficient force is exerted, the block will not slide down the ramp, because static friction holds it place and resists sliding. When objects have a high coefficient of friction, it means that a lot of force will be required to break through the force of static friction and create movement, while a low coefficient means that less force will need to be exerted. Most of the existing model-based friction compensation schemes use classical friction models, such as Coulomb and viscous friction. In applications with high precision positioning and with low velocity tracking, the results are not always satisfactory. A better description of the friction phenomena for low velocities and especially when crossing zero velocity is necessary. Friction is a natural phenomenon that is quite hard to model, and it is not yet completely understood. The classical friction models are described by static maps between velocity and friction force. Typical examples are different combinations of Coulomb friction, viscous friction, and Stribeck...
CHAPTER 4. FRICTION IDENTIFICATION

effect. The Coulomb friction model does not specify the friction force for zero velocity (Figure 4.1 (a)), see (Canudas de Wit et al., 1995) for more details. It may be zero or it can take on any value in the interval between Coulomb forces, depending on how the sign function is defined. Because of its simplicity, the Coulomb friction model has often been used for friction compensation (Friedland and Park, 1992). The term viscous friction is used for the friction force caused by the viscosity of lubricants (Olsson et al., 1998). Viscous friction is often combined with Coulomb friction as shown in Figure 4.1 (b). Short for static friction, stiction as illustrated in Figure 4.1(c) describes the friction force at rest (Morin, 1833). Static friction counteracts external forces below a certain level and thus keeps an object from moving. Strubeck observed in (Armstrong-Helouvry, 1993, 1990) shows that the friction force does not decrease discontinuously as it is illustrated in Figure 4.1(c), but the velocity dependence is continuous as shown in Figure 4.1(d). This type of friction is called Strubeck friction. The classical models explain neither hysteretic behavior when studying friction for non-stationary velocities nor variations in the break-away force with the experimental condition nor small displacements that occur at the contact interface during stiction.

The main disadvantage when using a model such as classical models, for simulations or control purposes, is the problem of detecting zero velocity. The model presented by Karnopp in (Richard and Cutkosky, 2002), was developed to overcome the problems with zero velocity detection, and to avoid switching between different state equations for sticking and sliding. The research work also shows how it can be used to create accurate and convincing displays of sliding friction, including pre-sliding displacement and stick-slip behavior. But, the drawback with the Karnopp model is that, the friction model is so strongly coupled with the rest of the system. The external force is an input to the model and this force is not always explicitly given. The model therefore has to be tailored for each configuration (Olsson et al., 1998).

To account for some of the observed dynamic friction phenomena, a classical model can be modified as proposed by (Armstrong-Hélovry et al., 1994). This friction model introduces temporal dependencies for stiction and Strubeck effect, but does not handle pre-sliding displacement. The pre-sliding is instead done by describing the sticking behavior by a separate equation. Some mechanism must then govern the switching between the model for sticking and the model for sliding.

Figure 4.1: Examples of static friction models. The friction force is given by a static function except possibly for zero velocity. a) shows Coulomb friction, b) Coulomb plus viscous friction. Stiction plus Coulomb and viscous friction is shown in c), and d) shows how the friction force may decrease continuously from the static friction level.
4.1.2 Dynamic friction models

A dynamic model describing the spring-like behavior during stiction was proposed by Dahl (Helmick and Messner, 2009; Padthe et al., 2006). The Dahl model is essentially Coulomb friction with a lag in the change of friction force when the direction of motion is changed. The model has many nice features, and it is also well understood theoretically. Questions such as existence and uniqueness of solutions and hysteresis effects were studied in (Bliman, 1992). The Dahl model does not however, include the St"{u}rbeck effect. An attempt to incorporate St"{u}rbeck effect into the Dahl model was done in (Bliman and Sorine, 1991) where the authors introduced a second order Dahl model using linear space invariant descriptions. The St"{u}rbeck effect in this model is only transient, however, after a velocity reversal and is not present in the steady-state friction characteristics. The Dahl model has been used for adaptive friction compensation (Barahanov and Ortega, 2000; Swevers et al., 2000; Lampaert et al., 2002) with improved performance as the results. There are also other models for dynamic friction. (Armstrong-Hélouvry et al., 1994) proposed a seven parameter model. This friction model does not combine the different friction phenomena, but is in fact one model for stiction and another for sliding friction. Another dynamic model suggested by (Swevers et al., 2000) makes an important modification to the LuGre model to allow accurate modeling both in the sliding and the pre-sliding regimes without the use of a switching function. The model incorporates a hysteresis function with nonlocal memory and arbitrary transition curves. (Lampaert et al., 2002) extended the research work done by (Swevers et al., 2000) to overcome discontinuity in the friction force which occurs during certain transitions in pre-sliding.

4.2 Worm-gears transmission

Worm-gears are transmission elements frequently used in electro-mechanical systems to transmit large torque and velocity ratios. Compared to the more conventional gears such as spur or bevel gears, in which a large number of gears and consequently a large space is required to achieve the desired gear ratios, worm-gears provide large torque and velocity ratios in a small space with a single gear set. Consequently Worm-gears allow one to adopt small, high velocity motors in applications involving slow motion (Yeh and Wu, 2005). The high gear ratio, (100 : 1) and the small leading angle \(^5\) result in a self-braking mechanism, reducing the use of the actuators to maintain a certain gear position, i.e., gearshift selection. Due to the self-locking properties of the Worm-gears, the free motion of the C-Lever without actuation is nearly impossible, and driving the joints without a proper compensation could result in a strange behaviour because of a high nonlinearity (stiction, St"{u}rbeck, Coulomb, and viscous frictions) in the joints. However, obtaining the friction model using multiple constant velocities is not feasible, as the workspace is constrained (±4 cm in x axis, and ±4 cm in y axis) and the Worm-gears are self-braking. Figure 4.2 depicts the Worm-gears transmission of the C-Lever mechanic. The worm is connected to the actuator (motor or drive) and is made of aluminium material. The worm gear segment is connected to the lever and is made of self-lubricating nylon material. Plastic gears have greater consistency and are easily molded to various shapes (e.g. worm gear segment). They are less expensive and are lighter in weight. Plastic gear are chemical and corrosion resistant and have quieter and smoother operation. They deflect to compensate for inaccuracies by absorbing tiny impacts of small tooth errors and gear misalignment. They offer more efficient drive geometry. Their self-lubrication property provide a maintenance free to the transmission. However, the dimension and properties of plastic gears change with temperature, moisture absorption and chemical exposure. They have reduced ability to operate at higher temperature and their teeth wear out more quickly under large stress or high gear ratio.

\(^5\)Angle between a tangent to the helix and a plane perpendicular to the axis of the Worm-gears.
4.3 Identification of the friction torque

The first step in identifying the friction’s parameters of the joint’s transmission is to obtain a data set of the friction torque and the joint velocity. A PD control law with acceleration feedforward and gravity torque compensation is defined to control the position of the joints during the identification procedure. Before defining the control law, a setpoint trajectories is designed using the following reference model to compute the reference position, velocity, and acceleration,

\[ J_m \ddot{q}_r + B_m \dot{q}_r + K_m q_r = r, \]  

(4.1)

where \( J_m \in \mathbb{R}^{2 \times 2} \), \( B_m \in \mathbb{R}^{2 \times 2} \), \( K_m \in \mathbb{R}^{2 \times 2} \) denote the inertia matrix, the damping coefficient matrix, the spring stiffness coefficient matrix of the reference model respectively. \( r \in \mathbb{R}^2 \) denotes the reference input vector, and \( q_r, \dot{q}_r, \) and \( \ddot{q}_r \) are the reference joint position, joint velocity, and joint acceleration respectively. Setting \( J_m \) as identity inertia matrix, \( B_m = \text{diag}\{2\zeta \omega_m\} \), and \( K_m = \text{diag}\{\omega_m^2\} \), and applying Laplace transformation to (4.1), leads to the following reference transfer function,

\[
H(s) = \begin{bmatrix}
\frac{\omega_m^2}{s^2 + 2\zeta \omega_m s + \omega_m^2} & 0 \\
0 & \frac{\omega_m^2}{s^2 + 2\zeta \omega_m s + \omega_m^2}
\end{bmatrix};
\]  

(4.2)

where \( \zeta \in \mathbb{R} \), \( \omega_m \in \mathbb{R} \) are the damping ratio and the natural frequencies of the reference model, respectively. To avoid differentiation error, the method illustrated in Figure 4.3 is applied to compute the reference position, velocity and acceleration.

Recalling the dynamic model of the C-Lever whose equation in Chapter 3 is given by

\[ M(q) \dddot{q} + C(q, \dot{q}) \ddot{q} + g(q) = \tau - \tau_f, \]

(4.3)

where \( M(q) \in \mathbb{R}^{2 \times 2} \) denotes the inertia matrix of the joints, \( C(q, \dot{q}) \dot{q} \in \mathbb{R}^2 \) represents the Coriolis and centrifugal forces, \( g(q) \in \mathbb{R}^2 \) denotes the gravitational force vector, \( \tau_f \in \mathbb{R}^2 \) denotes the friction torque vector, and \( \tau \in \mathbb{R}^2 \) is the input torque vector.

To avoid excitation of Coriolis and Centrifugal forces, a low reference velocity is used for identification, and the vector \( C(q, \dot{q}) \dot{q} \) is set to be equal to zero. Assuming the knowledge of the gravitational force, a PD control law with feedforward acceleration and gravity compensation illustrated in Figure 4.4 is defined as,

\[
\tau = K_{fa} \ddot{q}_r + K_P \dot{q} + K_D \dot{q} + g(q),
\]

(4.4)

where \( K_{fa} \in \mathbb{R}^{2 \times 2} \), \( K_P \in \mathbb{R}^{2 \times 2} \), \( K_D \in \mathbb{R}^{2 \times 2} \) denote the acceleration feedforward gain, the proportional control gain, and the derivative control gain matrices respectively. \( \ddot{q}_r, \dot{q}, \dot{q} \in \mathbb{R}^2 \) are the
4.3. IDENTIFICATION OF THE FRICTION TORQUE

The acceleration feedforward action is used for compensating the inertial torque of the joint’s dynamics. The proportional control gain is selected using the gravitational torque property in Appendix B.1 as,

\[ \lambda_{\text{min}} \{ K_P \} > k_g \quad \text{with} \quad k_g = \max g(q); \]

where \( \lambda_{\text{min}} \{ K_P \} \) denotes the smallest element of the proportional gain matrix, and \( k_g \) is the largest element of the gravitational torque vector. Furthermore, the derivative control gain is selected as,

\[ K_P \geq 10 \times K_D. \]

To compute the friction torque, the input torque in (4.4) is substituted in the dynamic model equation (4.3), and assuming that \( C(q, \dot{q})\ddot{q} = 0 \) yields to

\[ M(q)\ddot{q} = K_{fa}\ddot{q} + K_P\dot{q} + K_D\dot{q} - \tau_f. \]  \hspace{1cm} (4.5)
When rearranging (4.5), the friction torque is therefore obtained as

\[ \tau_f = K_{fa} \ddot{q}_r - M(q)\ddot{q} + K_p \dot{q} + K_D \dot{q}. \] (4.6)

4.4 Friction torque measurements

A sine-sweep trajectory defined in (4.7) is applied as the reference input. The frequency \( f_r \) is selected relatively low to avoid the excitation of the Coriolis and centrifugal terms, but also large enough to cover the pre-sliding and sliding modes of friction.

\[ r = A_r \sin(2\pi f_r t), \] (4.7)

where \( A_r \in \mathbb{R}^2, f_r \in \mathbb{R}^2 \) denote the amplitude and the frequency of the reference input respectively. The reference model’s parameters as defined in table 4.1 are used to perform these measurements.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Joint 1</th>
<th>Joint 2</th>
<th>units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_r )</td>
<td>15</td>
<td>15</td>
<td>Nm</td>
</tr>
<tr>
<td>( f_r )</td>
<td>0.5</td>
<td>0.5</td>
<td>Hz</td>
</tr>
<tr>
<td>( \omega_r )</td>
<td>10\pi</td>
<td>10\pi</td>
<td>rad/s</td>
</tr>
<tr>
<td>( \zeta )</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>

The control’s gains are also tuned to eliminate or to reduce the effects of other dynamics (e.g., inertial torque, gravitational torque, backlashes, etc.) on the tracking errors. The following gains are then used: \( K_P = \text{diag}\{10; 10\} \) [Nm/rad] with \( k_g = 0.707 \) Nm/rad, \( K_D = \text{diag}\{0.1; 0.1\} \) Nms/rad, and \( K_{fa} = \text{diag}\{0.005; 0.005\} \) Nms²/rad.

Figures 4.5 and 4.6 illustrate the references and positions of joints 1 and 2, respectively.

\[ \text{Displacement of joint 1} \]

![Figure 4.5: Displacement of joint 1](image-url)
4.4. Friction Torque Measurements

Figures 4.6 and 4.8 illustrate the tracking errors for both joints 1 and 2, respectively. The first observation made on these two figures is the dominant high stiction at velocity reversal or zero velocity (transition from forward to backward displacement) despite a smooth trajectory. Moreover, although the same characteristic is repeated over several measurements, the errors are not symmetric in both direction (forward and backward). A clear observation of a non-symmetric repeating errors is shown in Figure 4.8.

Figure 4.6: Displacement of joint 2

Figure 4.7: Tracking error of joint 1
Figures 4.8 and 4.10 illustrate the measured friction torques for both joints. The main observation from these results is the negative slope as the velocity increases, showing that the joint’s velocities remain smaller than the Stribeck velocity in the workspace. The small joint’s velocities are consequence of a high gear ratio of the Worm-gears transmission and a constrained workspace.
4.5. Friction model approximation

In (Weel, 2007), the parametric friction model was studied, based on the analytical understanding of the worm pairs transmission, whereafter a static friction model was derived. This approach is quiet interesting to study the full dynamical model of the friction. However, the approach appears to be very complex, and do not necessarily result in a good representation of the friction in the joints, due to several aspects like backlashes, mechanical play, or hysteresis that are not always included in the analysis.

4.5.1 Model selection

Amongst the abundant friction models from various literature (Barahanov and Ortega, 2000; Olsson et al., 1998), etc; the chosen model to approximate the measured friction torques is the static friction model with Stribeck velocity as described in equation 4.3 (Olsson et al., 1998). The static friction model with Stribeck velocity includes the nonlinear behavior of friction, without augmenting complexity to the model unlike Dahl, or Lureg model, etc. Moreover, all of the model parameters can be expressed as linear coefficients of the joints velocity. This fact is exploited to easy the estimation of the parameter’s values.

\[
\tau_f(\dot{q}) = \left[ \tau_c + (\tau_s - \tau_c) \exp \left( -\left| \frac{\dot{q}}{\dot{q}_s} \right|^{\delta} \right) + b|\dot{q}| \right] \text{sign}(\dot{q}),
\]

(4.8)

where \( \delta \in \mathbb{R}^2 \) is the adjusting parameter of the Stribeck curve and the constant vectors \( \tau_c, \tau_s, \dot{q}_s, b \in \mathbb{R}^2 \) represent the coulomb friction, the static friction, the Stribeck velocity, and the viscous friction coefficients, respectively. The major drawback of this friction model is the discontinuity at velocity reversal, or zero velocity due to the non smoothness of the \text{sign} function. The problem of discontinuity at zero velocity is resolved using a tangent function, \( \tanh \) with the smoothing parameter \( \lambda \) instead of the \text{sign} function. The modified equation is therefore written as,

\[
\tau_f(\dot{q}) = \left[ \tau_c + (\tau_s - \tau_c) \exp \left( -\left| \frac{\dot{q}}{\dot{q}_s} \right|^{\delta} \right) + b|\dot{q}| \right] \tanh(\lambda\dot{q}),
\]

(4.9)
4.5.2 Fitting the data to the friction model

We used the nonlinear least square method, called \textit{lsqcurvefit} from Matlab toolbox, to estimate the parameters of the model from the measurements data set (see Appendix E.2). Figure 4.11 illustrates the approximated friction model versus the non parametric friction model.

![Diagram](image)

Figure 4.11: Empirical static friction model
4.6 Friction model validation

The validation of the approximated friction model is carried out using a linear PD control plus static friction compensation scheme as depicted on figure 4.12. The input torque is defined as followed,

\[ \tau = \tau_f + K_P \tilde{q} + K_D \dot{\tilde{q}}; \quad (4.10) \]

where \( \tilde{q}, \dot{\tilde{q}} \) are the generalized position and velocity errors respectively. \( K_P \in \mathbb{R}^{2 \times 2} \) is the proportional control gain, \( K_D \in \mathbb{R}^{2 \times 2} \) is the derivative control gain, and \( \tau_f \in \mathbb{R}^2 \) is the vector of friction torques.

![Figure 4.12: Linear PD control + static friction compensation](image)

**Tracking errors.** Similar to the friction torque measurement procedure, the tracking errors are used in the model validation to observe the improved performance. Furthermore, the same reference trajectories are also used, with the following control’s gains;

\[ K_P = diag\{10; 10\} \text{ [Nm/rad]}, \quad K_D = diag\{0.1; 0.1\} \text{ [Nms/rad]}. \]

After some fine tuning, the approximated friction parameters are adjusted to the values shown in table 4.2. Figure 4.13 illustrates the tracking errors before friction compensation, and the improved tracking errors after friction compensation. The tracking errors after static friction compensation show an improved performance with a factor of 5 compared to the tracking errors without friction compensation.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>\textit{Joint}1</th>
<th>\textit{Joint}2</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_c )</td>
<td>0.24</td>
<td>0.2</td>
<td>Nm</td>
</tr>
<tr>
<td>( \tau_s )</td>
<td>1.1</td>
<td>.8533</td>
<td>Nm</td>
</tr>
<tr>
<td>( \dot{\tilde{q}}_s )</td>
<td>8.7266</td>
<td>3.4907</td>
<td>rad/s</td>
</tr>
<tr>
<td>( \dot{q} )</td>
<td>0.0005</td>
<td>0.0005</td>
<td>Nms/rad</td>
</tr>
<tr>
<td>( \delta )</td>
<td>0.2</td>
<td>0.12</td>
<td></td>
</tr>
<tr>
<td>( \lambda )</td>
<td>0.8</td>
<td>0.8</td>
<td></td>
</tr>
</tbody>
</table>
Figure 4.13: Tracking error
4.7 Discussion

In this work, we utilize an experimental approach to identify friction in the joint’s transmission. After collecting the friction torque data, we approximate the friction model parameters using a nonlinear least square approximation method in Matlab, and the modified static friction model in equation (4.5), where a smoothing parameter ($\lambda$) has been introduced to avoid discontinuity at zero velocity. We furthermore validate the approximated friction model using a PD control plus friction compensation, with the adjusted friction parameters presented in table 4.2. The tracking errors after friction compensation shows the correctness of the approximated friction model, with an improved tracking error five times better than the tracking error without friction compensation. The advantage of the approximate friction model is that, the friction torque computation is function of the joint’s velocities.
Chapter 5

Position and force-feedback control design

In this chapter, we define a control law that improves the performance of the C-Lever using the Euler-Lagrange equation of motion, and that enhances the haptic feeling using a force-feedback.

5.1 Control problem formulation

To define the control problem, we consider the dynamical model of the C-Lever and virtual environment force interaction as depicted in Figure 5.1. Unlike in robot manipulators, the trajectories of the end-effector (lever knob) are given by the human operator.

Figure 5.1: Representation of human interacting with the C-Lever
The Euler-Lagrange equation (3.4) in Chapter 3 is therefore modified to the following compact notation form,

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \tau_f = \tau - J^T(q)f_e, \]

where \( q \in \mathbb{R}^2 \) is the vector of joint displacements, \( \dot{q} \in \mathbb{R}^2 \) is the vector of joint velocities, \( M(q) \in \mathbb{R}^{2 \times 2} \) is a positive definite inertia matrix, \( C(q, \dot{q})\dot{q} \in \mathbb{R}^2 \) is the vector of centrifugal and Coriolis torques, \( g(q) \in \mathbb{R}^2 \) is the vector of gravitational torques, \( \tau \in \mathbb{R}^2 \) is the vector of actuation torques, \( J(q) \in \mathbb{R}^{2 \times 3} \) is the Jacobian matrix and \( f_e \in \mathbb{R}^3 \) is the interaction force vector with the human operator’s hand.

We are interested in the problem of defining the torque vector \( \tau \) to be applied to the joints to obtain the desired motion and the haptic feeling of equation (5.1) subject to actuators and workspace constraints. In particular, we want to apply the design methodology proposed in (Siciliano et al., 2009; Spong and Vidyasagar, 1989; Kelly et al., 2001; Åström and Hägglund, 2005) to design two control schemes,

1. **Inner-loop using inverse dynamics control to achieve the position control of equation (5.1)**

   The inverse dynamics problem consists of determining the joint torques \( \tau \) which are needed to generate the motion specified by the joint accelerations \( \ddot{q}(t) \), velocities \( \dot{q}(t) \), and positions \( q(t) \) once the possible end-effector forces \( f_h(t) \) are known (Siciliano et al., 2009). Solving the inverse dynamics problem is useful for manipulator trajectory planning and control algorithm implementation in the operational space. Once a joint trajectory is specified in terms of positions, velocities and accelerations (typically as a result of an inverse kinematics procedure), and if the end-effector forces are known, the inverse dynamics allows computation of the torques to be applied to the joints to obtain the desired motion.

2. **Outer-loop using force feedback control to enhance the feeling of the virtual environment of equation (5.1)**

   Force or haptic feedback can enhance a user’s feel of realism in a virtual environment simulation by conveying touch-related sensory information to the user. The user can be conveyed information about physical attributes of the simulated objects, like hardness, abruptness, or inertia through haptic feedback (e.g., gearshift patterns, etc). The outer-loop uses a force feedback control and a virtual admittance to compute the necessary set points as a function of the measured human force and the computed haptic force to provide the right haptic feeling to the operator.

Figure 5.2 illustrates the **Inner-loop/outer-loop** control configuration implemented for position control and force feedback control.

---

**Figure 5.2: Inner-loop/outer-loop control configuration**
5.2 Inner-loop controller

Several control techniques that stabilize arbitrary positions of robot manipulators can be found in literature. (Moreno-Valenzuela et al., 2007) for example, proposed a class of Output Feedback Tracking (OFT) for torque-saturated Robot Manipulators to tackle the torque-bounded problem, using Lyapunov stability and experimental evaluation. (Federico et al., June 2004), proposed a nonlinear anti-windup scheme to guarantee stability and local performance recovery on the Euler-Lagrange systems using computed torque control. (Cun-Qing et al., 200X) proposed a robust Nonlinear PID controllers for anti-windup design of robot manipulators with an uncertain jacobian matrix. But, the above control schemes are designed in the joint space of the manipulators and their setpoint trajectories are well defined in a trajectory planning scheme. In the case of the C-Lever, the desired trajectories are determined by the force’s interactions in the virtual environment (e.g., operator’s force and haptic feedback’s force). Therefore, the control law should not only guarantee stability and improve the performance, but should also generate the desired setpoint from the forces of the virtual environment.

We then use the inverse dynamics control design proposed in (Siciliano et al., 2009; Spong and Vidyasagar, 1989) to investigate on the design of the inner-loop controller. Moreover, we consider the dynamic model of the haptic device without any interaction with external forces in the inner-loop control design. Thereafter, we evaluate the designed controller using both simulink® for simulation and experimental setup, and we further compare the simulation results to experimental results. Therefore, the sampling frequency of the hardware (500 Hz), the encoder resolution (2π/2000 rad), the input torque resolution (12 bits), and the actuator saturation are included in simulink® environment.

5.2.1 Joint space inverse dynamics control

The approach that follows is founded on the idea to find a control vector \( \tau \), as a function of the system state, which is capable of realizing an input/output relationship of linear type; in other words, it is desired to perform an approximate linearization of the C-Lever dynamics obtained by means of a nonlinear state feedback. Consider the EL equation of motion of the joints (5.2), and assuming perfect knowledge of the model and friction,

\[
M(q)\ddot{q} + n(q, \dot{q}) = \tau, \tag{5.2}
\]

where for simplicity it has been set

\[
n(q, \dot{q}) = C(q, \dot{q})\dot{q} + g(q) + \tau_f(\dot{q}) \tag{5.3}
\]

The idea of the joint space inverse dynamics control is to seek a nonlinear feedback control law (5.4) when substitute in (5.2), results in a linear stable closed-loop system.

\[
\tau = f(q, \dot{q}, t). \tag{5.4}
\]

Taking the control \( \tau \) as a function of the joint’s state in the form

\[
\tau = M(q)v + n(q, \dot{q}), \tag{5.5}
\]

leads to a system double integrator described by,

\[
\ddot{q} = v \tag{5.6}
\]

where \( v \) represents a new input vector whose expression is yet to be determined, and \( M(q) \) is assumed known. The resulting control scheme is illustrated in Figure 5.3.
The nonlinear control law in (5.5) is called joint space inverse dynamics control because it is based on the computation of the joint’s equation of motion, whose state space form is presented as,

\[
\begin{bmatrix}
\dot{\mathbf{q}} \\
\ddot{\mathbf{q}}
\end{bmatrix} = 
\begin{bmatrix}
\dot{\mathbf{q}} \\
M^{-1}(\mathbf{q}) (-n(\mathbf{q}, \dot{\mathbf{q}}) + \tau)
\end{bmatrix},
\]

where \( \mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^2 \) represent the state positions and velocities of the joints respectively. \( M^{-1}(\mathbf{q}) \in \mathbb{R}^{2 \times 2} \) denotes the positive definite inverse inertia matrix, \( n(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^2 \) denotes the vector of nonlinear terms, and \( \tau \in \mathbb{R}^2 \) is the vector of input torque.

The system under control (5.5) is linear and decoupled with respect to the new input \( v \). In other words, the component \( v_1 \) or \( v_2 \) influence with a double integrator relationship, only the joint variable \( q_1 \) or \( q_2 \), independently of the motion of the other joint. In view of the choice (5.5), the control problem is reduced to that of finding a stabilizing linear control law \( v \). We then choose a PD control law,

\[
v = -K_P \mathbf{q} - K_D \dot{\mathbf{q}} + \mathbf{r},
\]

where \( \mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^2 \) represent the state positions and velocities of the joints respectively. \( K_P, K_D \in \mathbb{R}^{2 \times 2} \) denote the proportional and derivative gains matrices respectively. \( \mathbf{r} \in \mathbb{R}^2 \) denotes the input of a reference model which describes the new linear dynamics, \( v \in \mathbb{R}^2 \) is the linear control vector. Substituting (5.6) in (5.7) leads to the system of second order equations

\[
\ddot{\mathbf{q}} + K_P \mathbf{q} + K_D \dot{\mathbf{q}} = \mathbf{r},
\]

which, under the assumption of positive definite matrices \( K_P \) and \( K_D \), is asymptotically stable. Choosing \( K_P \) and \( K_D \) as diagonal matrices gives a linear and decoupled system. Given any reference trajectory \( q_r(t) \), tracking of this trajectory for the output \( q(t) \) is ensured by choosing

\[
\mathbf{r} = \ddot{q}_r + K_P q_r + K_D \dot{q}_r,
\]

where \( q_r, \dot{q}_r, \ddot{q}_r \in \mathbb{R}^2 \) represent the reference position, velocity and acceleration respectively. \( K_P \in \mathbb{R}^{2 \times 2} \) and \( K_D \in \mathbb{R}^{2 \times 2} \) denote the proportional and derivative gains matrices respectively.
5.2. INNER-LOOP CONTROLLER

$r \in \mathbb{R}^2$ denotes the input to the reference model. In fact, substituting (5.9) into (5.8) gives the homogeneous second-order differential equation (5.10). This homogeneous equation expresses the dynamics of the position error ($\tilde{q} = q - q_r$) while tracking the given trajectory. The position error occurs only if $\tilde{q}(0)$ and/or $\dot{\tilde{q}}(0)$ are different from zero and converges to zero with a speed depending on the matrices $K_P$ and $K_D$ chosen. Thus, the error equation is expressed as,

$$\ddot{\tilde{q}} + K_P \tilde{q} + K_D \dot{\tilde{q}} = 0,$$

(5.10)

where $\tilde{q} \in \mathbb{R}^2$, $\dot{\tilde{q}} \in \mathbb{R}^2$, $\ddot{\tilde{q}} \in \mathbb{R}^2$ represent the position, velocity, and acceleration states errors of the joints respectively. $K_P \in \mathbb{R}^{2 \times 2}$ and $K_D \in \mathbb{R}^{2 \times 2}$ denote the proportional and derivative positive definite gains matrices respectively. The control gains $K_P$ and $K_D$ are diagonal positive definite matrices, and are presented as followed,

$$K_P = \begin{bmatrix} \omega_{n1}^2 & 0 \\ 0 & \omega_{n2}^2 \end{bmatrix}, \quad K_D = \begin{bmatrix} 2\zeta_1 \omega_{n1} & 0 \\ 0 & 2\zeta_2 \omega_{n2} \end{bmatrix},$$

where $\omega_{n1}, \omega_{n2} \in \mathbb{R}$ denote the natural frequencies, and $\zeta_1, \zeta_2 \in \mathbb{R}$ denote the damping ratios. $\omega_{n1} = 2\pi f_{n1}$, and $\omega_{n2} = 2\pi f_{n2}$ with $f_{n1}, f_{n2} \in \mathbb{R}$ the respective frequencies in Hertz.

5.2.2 Evaluation of the joint space inverse dynamics control

The reference trajectories are computed using a reference model, and the control parameters for both simulation and experimental evaluations are chosen as followed in table 5.1. The position trajectories for joints 1 and 2 are illustrated in Figure 5.4 and 5.5.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Joint 1</th>
<th>Joint 2</th>
<th>units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{n1}$</td>
<td>10</td>
<td>0</td>
<td>Hz</td>
</tr>
<tr>
<td>$f_{n2}$</td>
<td>0</td>
<td>10</td>
<td>Hz</td>
</tr>
<tr>
<td>$\zeta_1$</td>
<td>0.707</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>$\zeta_2$</td>
<td>0</td>
<td>0.707</td>
<td>-</td>
</tr>
</tbody>
</table>
CHAPTER 5. POSITION AND FORCE-FEEDBACK CONTROL DESIGN

Figure 5.4: Trajectories of joint 1

Figure 5.5: Trajectories of joint 2
Simulation and experimental results

Figure 5.6(a) and 5.6(b) illustrate the tracking errors for joint 1 and joint 2, respectively. The tracking error from simulation and from experiments are similar on the first joint (Figure 5.6(a)). Moreover the error amplitude reaches the noise level where further improvement is not possible. But unexpectedly, the tracking error from experimental results on the second joint is different to the tracking error from simulation (Figure 5.6(b)). Moreover, the large peak of the error of joint 2 from experimental result are due to the position dependency friction, that causes a model parameter mismatched in the controller.

Figure 5.6: Illustration of tracking errors
Figure 5.7: Illustration of Joint’s torques
5.2. INNER-LOOP CONTROLLER

5.2.3 Operational space inverse dynamics control

In the joint space inverse dynamics control scheme, it is always assumed that the desired trajectory is available in terms of the time sequence of the values of joint position, velocity and acceleration. Accordingly, the error for the control scheme is expressed in the joint space.

As often pointed out, motion specifications are usually assigned in the operational space, and then an inverse kinematics algorithm has to be utilized to transform operational space references into the corresponding joint space references (Siciliano et al., 2009; Spong and Vidyasagar, 1989). The process of kinematic inversion has an increasing computational load when, besides inversion of direct kinematics, inversion of first-order and second-order differential kinematics is also required to transform the desired time history of end-effector position, velocity and acceleration into the corresponding quantities at the joint level.

The operational space inverse dynamics control approach consists of considering control schemes developed directly in the operational space. If the motion is specified in terms of operational space variables, the measured joint space variables can be transformed into the corresponding operational space variables through direct kinematics relations. Comparing the desired input with the reconstructed variables allows the design of feedback control loops where trajectory inversion is replaced with a suitable coordinate transformation embedded in the feedback loop.

Recall the C-Lever’s dynamic model in the form (5.11),

\[ M(q) \ddot{q} + n(q, \dot{q}) = \tau, \]

where \( n \) is given by (5.3). Under the assumption of a complete knowledge of the C-Lever’s dynamics and friction, the choice of the inverse dynamics linearizing control law

\[ \tau = M(q)v + n(q, \dot{q}), \]

leads to the system of double integrators,

\[ \ddot{q} = v. \]

The time differentiation of the workspace linear velocities leads to

\[ \ddot{x}_e = J_v(q)\ddot{q} + \dot{J}_v(q, \dot{q})\dot{q}, \]

which gives the relationship between the joint space accelerations and the operational space accelerations. The second order differential kinematics (5.14) can be inverted in terms of the joint acceleration

\[ \ddot{q} = J_v^\dagger(q) \left( \ddot{x}_e - \dot{J}_v(q, \dot{q})\dot{q} \right), \]

The numerical integration of (5.15) to reconstruct the joint velocities and positions would unavoidably lead to a drift of the solution; therefore, it is worth to consider the position error \( \ddot{x} = x_e - x_d \), and its second order time derivative \( \dddot{x} = \ddot{x}_e - \ddot{x}_d \) which in view of (5.14), yields

\[ \dddot{x} = \ddot{x}_e - J_v(q)\ddot{q} - \dot{J}_v(q, \dot{q})\dot{q}. \]

The choice of the control law is therefore,

\[ v = J_v^\dagger(q)(\ddot{x}_d + K_D \dddot{x} + K_P \dddot{x} - \dot{J}_v(q, \dot{q})\dot{q}), \]

where \( \ddot{x}_d \in \mathbb{R}^3 \) denotes the workspace desired accelerations, \( \dot{q} \in \mathbb{R}^2 \) is the joint velocities, \( J_v^\dagger \in \mathbb{R}^{3 \times 3} \) denotes the pseudo-inverse of the Jacobian and \( \dot{J}_v \in \mathbb{R}^{3 \times 2} \) is the time derivative Jacobian of linear velocities. \( K_P, K_D \in \mathbb{R}^{3 \times 3} \) the proportional and derivative positive definite
gain matrices. In fact, substituting (5.17) and (5.13) into (5.16) leads to the equivalent linear error system

\[
\ddot{\tilde{x}} + K_P \tilde{x} + K_D \dot{\tilde{x}} = 0,
\]

which is asymptotically stable. The error tends to zero along the trajectories with a convergence speed depending on the choice of the matrices \(K_P\) and \(K_D\). Moreover, the dynamics in the workspace are decoupled by the stabilizing control gains \(K_P\) and \(K_D\), which are diagonal positive definite matrices, and are presented as followed,

\[
K_P = \begin{bmatrix}
\omega_{n1}^2 & 0 & 0 \\
0 & \omega_{n2}^2 & 0 \\
0 & 0 & 0
\end{bmatrix},
\]

\[
K_D = \begin{bmatrix}
2\zeta_1\omega_{n1} & 0 & 0 \\
0 & 2\zeta_2\omega_{n2} & 0 \\
0 & 0 & 0
\end{bmatrix},
\]

where \(\omega_{n1}, \omega_{n2} \in \mathbb{R}\) denote the natural frequencies, and \(\zeta_1, \zeta_2 \in \mathbb{R}\) denote the damping ratios. \(\omega_{n1} = 2\pi f_{n1}\), and \(\omega_{n2} = 2\pi f_{n2}\) with \(f_{n1}, f_{n2} \in \mathbb{R}\) the respective bandwidth control frequencies. The resulting inverse dynamics control scheme is reported in figure 5.8.

![Block scheme of operational space inverse dynamics control](image)

**Figure 5.8:** Block scheme of operational space inverse dynamics control

### 5.2.4 Evaluation of the operational space inverse dynamics control

Similar to the joint space inverse dynamics control scheme, this control scheme evaluation is carried out in simulation environment and on the experimental setup. A sine-sweep is also used to conduct the evaluation. Moreover, the reference trajectories are computed using a reference model, and the control parameters for both simulation and experimental evaluations are chosen as followed in table 5.3.
Table 5.2: Operational space: Control’s parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>x-axis</th>
<th>y-axis</th>
<th>z-axis</th>
<th>units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{n1}$</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>Hz</td>
</tr>
<tr>
<td>$f_{n2}$</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>Hz</td>
</tr>
<tr>
<td>$f_{n3}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Hz</td>
</tr>
<tr>
<td>$\zeta_1$</td>
<td>0.707</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>$\zeta_2$</td>
<td>0</td>
<td>0.707</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>$\zeta_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

(a) Displacements : x axis

(b) Displacements : y axis

Figure 5.9: Workspace trajectories
Simulation and experimental results

Figures 5.10(a) and 5.10(b) illustrate the position’s errors for both x axis and y axis, respectively. Unlike in simulation, the position’s errors from the experimental results show the presence of some friction dynamics, particularly on y axis. However, the position errors are not affected by noises (e.g., input torque resolution, or encoder’s resolution), due to the used of kinematic transformation. Moreover, the control bandwidth and the trajectories in the workspace are determined, and the control efforts are similar to those obtained from the joint space inverse dynamics control scheme as illustrated in Figures 5.11(a) and 5.11(b). Figures 5.12(a) and 5.12(b) illustrate the displacements of joint 1 and 2 respectively.

![Position error x axis](image1)

(a) $x - x_d$

![Position error y axis](image2)

(b) $y - y_d$

Figure 5.10: Workspace’s position errors
5.2. INNER-LOOP CONTROLLER

Figure 5.11: Joint’s torques
Displacement of joint 1

![Graph](image1)

(a) Experiments: $q_1$

Displacement of joint 2

![Graph](image2)

(b) Experiments: $q_2$

Figure 5.12: Experimental’s results: Joint’s displacements
5.2.5 Operational space control vs joint space control

Figures 5.13 and 5.14 illustrate the position errors for both operational space control and joint space control, respectively. In contrast to the joint space control, the tracking error of the operational space control does not contain measurement noises, due to the kinematic transformation. The stabilizing linear control is implemented in the workspace, as compared to the joint space control. The control bandwidth of the operational space control is defined for the workspace dynamics. Moreover, the workspace trajectories are known, and the dynamics are also decoupled using the stabilizing linear control gain matrices. Nonetheless, both control schemes are sensitive to unmodeled dynamics (e.g., mechanical play, backlashes, position dependency friction, etc.), that lead to parameter’s mismatched in the controller. The consequence of a position dependency friction is clearly observed on the workspace’s position error (Figure 5.13). Throughout this analysis, the operational space inverse dynamics control is further used to implement the haptic patterns.

![Position error x axis](image1)

Figure 5.13: Workspace’s position error

![Tracking error of joint 1](image2)

Figure 5.14: Joint space’s position error
5.3 Outer-loop controller

Figure 5.15 illustrates the outer-loop block scheme with a force feedback control and virtual admittance. $f_h$ and $f_d$ denote the human arm force, and the desired force from the haptic patterns respectively. The haptic feeling is enhanced by the force-feedback using a PID controller with anti-windup, the desired setpoint $x_d$ are derived using the virtual admittance $Z^{-1}(s)$.

![Diagram of outer-loop controller with force feedback control](image)

5.3.1 Virtual admittance

The setpoint trajectories of the Inner-loop controller (reference position, velocity, and acceleration) are derived using the virtual admittance $Z^{-1}(s)$. Therefore, $Z^{-1}(s)$ is the transfer function between the set forces of the virtual environment and the setpoint trajectories of the Inner-loop or position controller. We derive the virtual admittance using the following equation,

$$M_e \ddot{x}_d + B_e \dot{x}_d + K_e x_d = f_e, \quad (5.19)$$

where $M_e \in \mathbb{R}^{3 \times 3}$, $B_e \in \mathbb{R}^{3 \times 3}$, $K_e \in \mathbb{R}^{3 \times 3}$ denote the Mass matrix, the damping coefficient matrix, the spring stiffness coefficient matrix of the virtual environment. $f_e \in \mathbb{R}^3$ denotes the virtual environment vector forces, and $x_d$, $\dot{x}_d$, and $\ddot{x}_d$ are the desired position, velocity, and acceleration respectively.

To simplify the model, we set $M_e$ as identity mass matrix, $B_e = diag\{2\zeta\omega_e\}$, and $K_e = diag\{\omega_e^2\}$, and using Laplace transformation leads to the following transfer function matrix,

$$Z^{-1}(s) = \begin{bmatrix}
\frac{\omega_1^2}{s^2 + 2\zeta\omega_1 + \omega_1^2} & 0 & 0 \\
0 & \frac{\omega_2^2}{s^2 + 2\zeta\omega_2 + \omega_2^2} & 0 \\
0 & 0 & \frac{\omega_3^2}{s^2 + 2\zeta\omega_3 + \omega_3^2}
\end{bmatrix}; \quad (5.20)$$

where $\zeta \in \mathbb{R}$, $\omega_{e1}$, $\omega_{e2}$, $\omega_{e3} \in \mathbb{R}$ are the damping ratio and the natural frequencies of the virtual admittance, respectively.
5.3.2 Force feedback control using PID with anti-windup

Conventionally, control of most haptic interfaces is achieved through an open loop or closed loop impedance controller. However, conversely to impedance controlled interface, the C-Lever is an admittance controlled interface. The device measures the force that the human operator exerts on the lever knob, and reacts with motion (acceleration, velocity, position). Nonetheless, beside the operator force intended to shift gear, there exists some disturbance forces. The PID control is therefore used to correct the set forces at the lever knob, before the virtual admittance and the inner-loop control. The feedback loop around the integral action prevents the integrator windup, as a consequence of actuator saturation when the operator applies large forces on the lever knob. It is advantageous not to reset the integrator instantaneously, but dynamically with a time constant $T_i$ (Åström and Hågglund, 2005). The set forces at the end-effector (lever knob) are therefore derived as follows,

$$f_e = K_f \left[ \tilde{f} + T_d \dot{\tilde{f}} + \left( \frac{1}{T_i} - \frac{\tilde{f}_s}{T_t} \right) \int_0^t \tilde{f} \, d\tau \right],$$  \hspace{1cm} (5.21)

where $\tilde{f} = f_d - f_h$ is the force error, and $\tilde{f}_s = f_{lim} + \frac{KT_t}{T_i} \tilde{f}$ with $f_{lim}$ the maximum value of $f_e$. $K_f \in \mathbb{R}^{3 \times 3}$ denotes the force control gain matrix, $T_d \in \mathbb{R}$ denotes the derivative time constant, $T_i \in \mathbb{R}$ denotes the integral time constant and $T_t \in \mathbb{R}$ denotes the tracking time constant. The rate at which the controller output is reset is governed by the feedback gain, $1/T_t$, where $T_t$ determines how quickly the integral is reset. The tracking time constant $T_t$ should therefore be larger than $T_d$, and smaller than $T_i$. A rule of thumb that has been suggested is to choose $T_t = \sqrt{T_i T_d}$ (Åström and Hågglund, 2005).

5.4 Free motion and virtual wall

The free motion and virtual wall are implemented to evaluate the combined Inner-loop controller using operational space inverse dynamics, and the outer-loop controller using the force-feedback control. The spring-damper are used to model the virtual wall as followed,

$$f_d = \begin{bmatrix} K_w \left( 1 + \frac{1}{\sigma_T} T_{wd} \right) (x - \text{sat}(x)) \\ K_w \left( 1 + \frac{1}{\sigma_T} T_{wd} \right) (y - \text{sat}(y)) \\ 0 \end{bmatrix},$$  \hspace{1cm} (5.22)

where $K_w \in \mathbb{R}$, $T_{wd} \in \mathbb{R}$ denote the spring coefficient and the damping coefficient of the virtual wall respectively, and $\frac{1}{\sigma_T}$ denoting the time derivative. $\text{sat}(x)$ and $\text{sat}(y)$ denote the virtual limits of the workspace in x-axis and y-axis respectively.
Experimental results

For the experimental evaluation, the virtual limits are taken as, \( \text{sat}(x) = \text{sat}(y) = [-3, 3] \) cm, the force control gain matrix \( K_f \) is set as a unity matrix and the parameters of table 5.3 are selected for the PID controller and the virtual wall. The parameters of the virtual wall \( K_w \), and \( T_{wd} \) are remarkably small, due to the self-braking properties of the Worm-gears that increase the feeling of the virtual wall, once the lever knob reaches the virtual limits.

Table 5.3: Free motion and virtual wall: Force-feedback’s parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>x-axis</th>
<th>y-axis</th>
<th>z-axis</th>
<th>units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_w )</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>N/m</td>
</tr>
<tr>
<td>( T_{wd} )</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>( T_i )</td>
<td>1000</td>
<td>1000</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>( T_t )</td>
<td>500</td>
<td>500</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>( T_d )</td>
<td>0.005</td>
<td>0.005</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

To achieve the same stiffness between the haptic device and the virtual environment, the natural frequencies of the virtual admittance and the inner-loop controller are chosen identical. A bandwidth of 5 Hz is then selected to carry out the experiment. The main observation during the free motion is the changing behaviour of the trajectories, as the operator’s hand stiffness changes. The bandwidth is then reduced to 3 Hz to avoid the excitation of unmodeled dynamics (e.g., Mechanical play, gear misalignment, etc).

Figure 5.16 illustrates the trajectories of the lever knob obtained during the free motion experiments. The borders of the spanned workspace indicate the presence of the virtual wall in x-axis and y-axis. Figures 5.17(a), and 5.17(b) illustrate also the same trajectories in 2D, where the forces are plotted versus the positions in x-axis and y-axis. The operator’s forces are larger than the desired forces inside the workspace during the free motion, and the desired forces become larger or equal to the operator’s forces at the borders of the virtual wall, giving the sensation of hitting a wall.

![Figure 5.16: Free motion trajectories in the workspace](image)
5.4. FREE MOTION AND VIRTUAL WALL

Operator’s force and desired force in $x$–axis

(a) Forces versus $x$–axis

Operator’s force and desired force in $y$–axis

(b) Forces versus $y$–axis

Figure 5.17: Free motion in 2D: Forces versus positions
5.5 Discussion

Through this chapter the control problem of equation (5.2) with actuator’s inputs constraint and workspace constraint is investigated using two control schemes, the inner-loop controller to achieve a position control of the haptic device, and the outer-loop controller to enhance the haptic patterns feeling.

In contrast to the joint space inverse dynamics control scheme, the operational space inverse dynamics controller is less sensitive to measurement noises. The control bandwidth and the trajectories are well defined in the workspace. Nevertheless, the controller lacks of robustness against unmodeled dynamics and requires more computational efforts than the joint space inverse dynamics control. The complete control scheme (Inner-loop/outer-loop control) is evaluated using the free motion and the virtual wall, and a bandwidth of 3 Hz is chosen to further carry out the haptic implementation.
Chapter 6

Haptic implementation

The implementation of the virtual environment (automatic gear shift) is achieved using the haptic patterns and the force feedback control or outer loop control before the position control law. The complete control scheme used for implementing the haptic patterns in this research work is illustrated in figure 6.1. The haptic patterns and the force feedback controller provide the virtual sensation (e.g. gearshift feeling) to the human operator, and the virtual admittance is used to generate the reference trajectories from the force-feedback controller.

Figure 6.1: Control scheme for the haptic implementation (Adapted from (García-Canseco et al., 2010a).)

6.1 Haptic patterns

The haptic patterns are designed based on generalized sigmoid functions (García-Canseco et al., 2010b). The two main advantages of this technique are on one hand, the smoothness of the force field, due to continuous first and higher order derivatives, and on the other hand, the easy
6.2 The operator’s force

The operator’s force is processed via some piezo-electric force sensors, called FlexiForce force sensors. As shown on figure C.1, the sensors are ultra-thin and flexible printed circuit, which can be easily integrated into most applications. With their paper-thin construction, flexibility and force measurement ability, the FlexiForce force sensors can measure force between almost any two surfaces with a force range of 0 to 4N and are durable enough to stand up to most environments. But some induced noises caused by fluctuation in the power supply, quality of the hardware (poor soldering or wiring), bits error in the data acquisition, etc, affect the quality of the measured output voltage force. A filter is therefore designed to improve the signal to noise ratio of the measured force signal.

Denoising the force signal

A digital filtering technique using a direct-form realization of FIR with an Exponential Weighted Moving Average coefficients is implemented in software environment, to denoise the force signal. This filtering structure is based on the finite impulse response (FIR) technique, which is known to be causal and has a linear phase delay. Details about the design of the filter are presented in Appendix C.
6.3 Experimental results

The implementation of the designed gearshift patterns in Figure 6.2, the inner control loop and the outer control loop is achieved using Matlab Simulink® and dSpace® software that come along with the DS1102 controller board. An overview of the implementation process is depicted in figure 6.4.

![Software implementation process diagram]

**Automatic gearshift patterns**

Figure 6.5 illustrates the result of the automatic gearshift patterns implementation in the virtual environment. "D" denotes the Drive mode, "R" denotes the Reverse mode, and "N" denotes the Neutral. The attraction forces of the patterns increase as the lever knob is nearer a certain pattern (R, or N, or D), providing the sensation of gearshift engagement. Moreover, these forces also give the sensation of a gearshift bumps when the operator moves the lever away from the engaging position. If the operator releases the lever knob between two patterns, it falls back to the closest pattern. Furthermore, there is no unique trajectory to move the lever from one gear selection to another, providing flexibility in the manoeuvre during the shifting.

6.4 Discussion

The implementation goal of this project was to test a various range of haptic patterns. However, the automatic gearshift has been the only successful patterns tested. After some investigations, it was clearly noticed that the available workspace is not large enough to implement other haptic patterns (e.g., 5-gear + reverse patterns, H patterns, Y patterns, etc). Moreover, the delay in the force sensor activation, the mechanical play and the gear misalignments limit the performance of the implemented patterns. The processing time get slower, and the system runs out of memory (memory time out) when a five gears haptic patterns is implemented.
CHAPTER 6. HAPTIC IMPLEMENTATION

Figure 6.5: Virtual environment: automatic gearshift patterns
Conclusions and Recommendations

Conclusions about the research work are presented in this chapter, followed with some recommendations for future works.

7.1 Conclusions

In this research work, we study the kinematics and the dynamic model of the C-Lever using Denavit-Hartenberg (D-H) convention and Euler-Lagrange formulation respectively; but we further define an alternative set of kinematics using rotation matrix to avoid singularity found in the kinematic model derived by the D-H convention. To include friction in the dynamic model, we carry out a friction identification method using a sine-sweep trajectory and a designed controller that eliminates the gravitational and inertial torques during measurements. Moreover, due to the large stiction of the worm-pair transmission, we select a static friction model with Stribeck velocity, we utilize a least square method to estimate the parameters of the model using the data set of the measured friction torque. Then, we modify the friction model modified by replacing the sign function with a smooth function to eliminate discontinuity at velocity reversal. The modified static friction model with Stribeck velocity contains the nonlinear characteristics of friction, but remains simple and easy to implement compare to the dynamic friction models. Furthermore a model validation, followed the friction model parameter’s estimation, using a PD controller plus friction compensation on the experimental setup. After formulation of the control problem, we investigate two control schemes. The inner-loop control scheme using inverse dynamics for position control and the outer-loop control scheme using PID with anti-windup for force feedback control. The aim of inverse dynamics control is to improve the control bandwidth, with more feedback while keeping the control efforts within the actuator limits, and to derive the joint’s trajectories systematically using the computed control torques. We therefore study the inverse dynamics control in the joint space and the operational space. In contrast to the joint space inverse dynamics control, the operational space inverse dynamics control scheme is less sensitive to measurement noises, as illustrated by the tracking errors. The control bandwidth is well defined in the operational space, with similar control efforts as the joint space inverse dynamics control. Nevertheless, the inverse dynamics control does not handle unmodeled dynamics, and parameter’s uncertainty as observe in the tracking errors of both designed control schemes. Furthermore, the operational space inverse dynamics is complex, and requires more memory space and a fast computation time. Followed the inner-loop controller, we design and implement a force feedback control using PID with anti-windup and a virtual admittance in the outer-loop controller. The aim of the outer-loop controller is to enhance the haptic feeling and to generate the setpoint trajectories. We thereafter test the combined control scheme (Force feedback using PID with anti-windup, and operational space inverse dynamics), using free motion and virtual wall implementation. Prior to the test, we choose the virtual admittance and inner-loop control bandwidth identical to
achieve the same stiffness between the device and the virtual environment. Furthermore, the low values of the virtual wall parameters show the advantage of the self-locking properties of the Worm-gear transmission, in reducing the control efforts.

The influence of the human hand stiffness on the behaviour of the trajectories, when a large bandwidth is used, main observation from the free motion evaluation. We therefore carry out the implementation of the haptic patterns using a lower bandwidth for both the virtual admittance and inner-loop control bandwidth.

To complete the research work, we implement and test some haptic patterns (e.g., Automatic gearshift, etc). Throughout the implementation of these haptic patterns, we investigate problems related to the setup, the design of the controller, and the design of the haptic patterns. The results of these investigations are discussed in the recommendations section with some suggestions for future research works.

7.2 Recommendations

The dynamic model of the haptic device was based on the rigid body concept, and the design of the inner-loop controller is based on the assumption of a complete knowledge of the dynamic model. Although friction is identified and compensated, the dynamics caused by mechanical play, backlashes, and gear misalignment are not controlled. Consequently, the inner-loop controller lacks of robustness, the control bandwidth is therefore reduced to improve the stability of the trajectories in the workspace. Moreover, the automatic gearshift patterns is the only successful haptic implementation. The implementation of other haptic patterns (e.g., five gears shifting patterns, H or Y patterns, etc) is not possible, due to the constraint workspace, the memory space and the processor speed (500 Hz). Besides, the delay during activation of the force sensors causes the stalling of actuators during fast gear shifting. The torque resolution (12 bits) and the encoder resolution $\frac{2\pi}{2000}$ limit the achieved performance by increasing the noise level on the control torque. The following suggestions are therefore underlined for future works,

- **Haptic device**
  Redesign the setup device, to eliminate the excessive mechanical play, gear misalignments, and position dependent friction. Moreover, increase the workspace to allow more flexibility to implement a two degrees-of-freedom haptic patterns. Furthermore, design a mechanical pre-activation system for the force sensors, to avoid the stalling of actuators during fast gear shifting.

- **Robustness of the control design**
  Design and implement a robust control loop in the outer-loop control scheme, to handle unmodeled dynamics. Alternatively, study other type of continuous controllers that can handle constrained actuators, workspace and unmodeled dynamics.

- **Haptic patterns design**
  Investigate the human arm model with the C-Lever for simulation purposes, of several haptic patterns with the combined haptic device model before implementation. Furthermore, design an optimization algorithm to automatically tune the haptic pattern’s parameters to adapt the force-feedback to the user’s efforts.

- **Software/Hardware environments**
  Upgrade the hardware unit, for more memory space, fast processing time to easy the implementation of the model-based control, haptic patterns, and a graphical user interface. Upgrade the operating system and the design software package to acquire more flexibility for the design environment, e.g., use a real time operating system (Linux), Matlab simulink® toolbox, and C program, etc.

- **Graphical user interface**
  Design a friendly graphical user interface for visualization of the haptic patterns in real time during testing and for multiple haptic patterns choices from the operator.
Bibliography


H. P. Buchs, H. Versmold, and I. Neunzig.


S. Burkhardt, R. Ehrmaier, S. Furst, and J. Neuner.


Appendix A

Specification of the haptic setup

Hardware:
- Miniature Analog DC Servo Amplifier for DC brush motors: VIOLIN 15/60
- MAXON DC brush motor with permanent magnet, type Pg 4520
- dSpace© 1102 data acquisition system
- FlexiForce sensors , model A201, force range 0-1 lb (4.4 N)
- PC: Intel mainboard, model 486DX-2/66 MHz,16 MB of RAM

Software:
- OS: Microsoft windows 98
- dSpace© control desk developer version 1.2
- Analysis and simulation: Matlab 5.3 and simulink© 3.0
APPENDIX A. SPECIFICATION OF THE HAPTIC SETUP

Figure A.1: Experimental setup environment
Appendix B

Model properties

B.1 Gravitational forces property used for control gain selection

The gravitational torques vector \( g(q) \) a manipulator (e.g., C.Lever), of dimension \( n \times 1 \), depends only on the joint positions \( q \). The vector \( g(q) \) is continuous and therefore bounded for each bound \( q \). Moreover, \( g(q) \) also satisfies the following.

1. The vector \( g(q) \) and the velocity \( \dot{q} \) are correlated as

\[
\int_0^T g(q(t))^T \dot{q}(t) \, dt = \mathcal{U}(q(T)) - \mathcal{U}(q(0))
\]

for all \( T \in \mathbb{R}_+ \)

2. For the manipulator having only revolute joints there exists a number \( k_{U} \) such that

\[
\int_0^T g(q(t))^T \dot{q}(t) \, dt + \mathcal{U}(q(0)) \geq k_{U}
\]

for all \( T \in \mathbb{R}_+ \) and where \( k_{U} = \min_{q} \{ \mathcal{U}(q) \} \).

3. For the manipulator having only revolute joints, the vector \( g(q) \) is Lipschitz, that is, there exists a constant \( k_{g} > 0 \) such that

\[
\| g(x) - g(y) \| \leq k_{g} \| x - y \|
\]

for all \( x, y \in \mathbb{R}^2 \). A simple way to compute \( k_{g} \) is by evaluating its partial derivative

\[
k_{g} \geq 2 \left( \max_{i,j,q} \left| \frac{\partial g_i(q)}{\partial q} \right| \right)
\]

Furthermore, \( k_{g} \) satisfies

\[
k_{g} \geq \| \frac{\partial g(q)}{\partial q} \| \geq \lambda_{Max} \{ \frac{\partial g(q)}{\partial q} \}.
\]

4. For the manipulator having only revolute joints there exists a constant \( k' \) such that

\[
\| g(q) \| \leq k'
\]

for all \( q \in \mathbb{R}^2 \)
APPENDIX B. MODEL PROPERTIES

B.2 Inertia matrix elements

\[ M(q) = \begin{bmatrix} M(1, 1) & M(1, 2) \\ M(2, 1) & M(2, 2) \end{bmatrix}, \]

where

\[ M(1, 1) = \frac{1}{2} \left( 2J_{1y} + J_{2x} + J_{2z} + l^2 m_2 + 2m_1 r^2 r_1 z + 2m_1 r^2_1 z \right) \]
\[ + \frac{1}{2} \left( 2lm_2 r_2 x + m_2 r^2_2 x + m_2 r^2_2 y - J_2 x + J_2 z + m_2 (l^2 + 2lr_{2x}) \right) \]
\[ + \frac{1}{2} \left( (r^2_2 x - r^2_2 z) C_{2q_2} + 2m_2 (l + r_{2x}) r_{2z} S_{2q_2} \right) \]

\[ M(2, 1) = m_2 r_2 y \left( r_{2z} \cos(q_2) - (l + r_{2x}) \sin(q_2) \right) \]

\[ M(1, 2) = m_2 r_2 y \left( r_{2z} \cos(q_2) - (l + r_{2x}) \sin(q_2) \right) \]

\[ M(2, 2) = J_{2y} + m_2 (l^2 + 2lr_{2x} + r^2_{2x} + r^2_{2z}) \]

B.3 Coriolis and centrifugal forces matrix elements

\[ C(q, \dot{q}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}, \]

where

\[ C_{11} = \frac{1}{2} \left( -2m_2 (l + r_{2x}) r_{2z} \cos(2q_2) + (-J_{2x} + J_{2z} + m_2 (l^2 + 2lr_{2x} + r^2_{2x} - r^2_{2z}) \sin(2q_2) \right) \]

\[ C_{12} = \frac{1}{2} \left( 2m_2 (l + r_{2x}) r_{2z} \cos(2q_2) + (J_{2x} - J_{2z} - m_2 (l^2 + 2lr_{2x} + r^2_{2x} - r^2_{2z}) \sin(2q_2) \right) \]

\[ C_{21} = \frac{1}{2} \left( 2m_2 (l + r_{2x}) r_{2z} \cos(2q_2) + (J_{2x} - J_{2z} - m_2 (l^2 + 2lr_{2x} + r^2_{2x} - r^2_{2z}) \sin(2q_2) \right) \]

\[ C_{22} = -m_2 r_2 y ((l + r_{2x}) \cos(q_2) + r_{2z} \sin(q_2)) \]

B.4 Gravitational forces vector elements

\[ g(q) = \begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix}, \]

where

\[ g_1(q) = 0 \]
\[ g_2(q) = -gm_2 ((l + r_{2x}) \cos(q_2) + r_{2z} \sin(q_2)) \]
B.5  Haptic device’s parameters

The following haptic device’s parameters (table B.1) are computed using Pro−Engineer®. Pro−Engineer® is a parametric, integrated 3D CAD/CAM/CAE solution created by Parametric Technology Corporation (PTC).

Table B.1: haptic device’s parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>values</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>m_1</td>
<td>0.6 kg</td>
</tr>
<tr>
<td></td>
<td>m_2</td>
<td>0.4 kg</td>
</tr>
<tr>
<td></td>
<td>j_{1x}</td>
<td>4.45 \times 10^{-5} Nms^2/rad</td>
</tr>
<tr>
<td></td>
<td>j_{1y}</td>
<td>4.29 \times 10^{-5} Nms^2/rad</td>
</tr>
<tr>
<td></td>
<td>j_{1z}</td>
<td>4 \times 10^{-5} Nms^2/rad</td>
</tr>
<tr>
<td></td>
<td>j_{2x}</td>
<td>5.77 \times 10^{-5} Nms^2/rad</td>
</tr>
<tr>
<td></td>
<td>j_{2y}</td>
<td>4.74 \times 10^{-5} Nms^2/rad</td>
</tr>
<tr>
<td></td>
<td>j_{2z}</td>
<td>3.55 \times 10^{-5} Nms^2/rad</td>
</tr>
<tr>
<td></td>
<td>r_{1x}</td>
<td>3 \times 10^{-2} m</td>
</tr>
<tr>
<td></td>
<td>r_{1y}</td>
<td>2.73 \times 10^{-2} m</td>
</tr>
<tr>
<td></td>
<td>r_{1z}</td>
<td>2.35 \times 10^{-3} m</td>
</tr>
<tr>
<td></td>
<td>r_{2x}</td>
<td>4.64 \times 10^{-2} m</td>
</tr>
<tr>
<td></td>
<td>r_{2y}</td>
<td>4.55 \times 10^{-2} m</td>
</tr>
<tr>
<td></td>
<td>r_{2z}</td>
<td>1.39 \times 10^{-2} m</td>
</tr>
<tr>
<td></td>
<td>\ell</td>
<td>15 \times 10^{-2} m</td>
</tr>
</tbody>
</table>

B.6  Robotica data file

We have used Robotica® to compute Kinematics, Jacobian, and the equations of motion, with the following input data file. Robotica® is a program developed by (Nethery and Spong, 1994) to compute the kinematics and the dynamic equations of motion for robot manipulators in mathematica.

DOF = 2

The Denavit-Hartenberg parameters:

\textbf{joint1 = revolute}
a1 = 0  
alpha1 = \text{Pi}/2  
d1 = 0  
theta1 = q1

\textbf{joint2 = revolute}
a2 = 1  
alpha2 = \text{Pi}/2  
d1 = 0  
theta2 = q2

The dynamics information:

DYNAMICS
G1 = \{0,0,0\}
G2 = \{0,0,-g\}

mass1 = m1
center of mass = \{r1x,r1y,r1z\}
inertia matrix = \{j1x,0,0,j1y,0,j1z\}

mass2 = m2
center of mass = \{r2x,r2y,r2z\}
inertia matrix = \{j2x,0,0,j2y,0,j2z\}
Appendix C

Force sensor’s filter design

The details about the choice and the design of the force sensor’s filter are presented in the following sections.

C.1 Force sensors

The operator’s force is process via some piezo-electric force sensors, called FlexiForce force sensors. As shown on figure C.1, the sensors are ultra-thin and flexible printed circuit, which can be easily integrated into most applications. With their paper-thin construction, flexibility and force measurement ability, the FlexiForce force sensors can measure force between almost any two surfaces with a force range of 0 to 4N and are durable enough to stand up to most environments. But some induced noises caused by fluctuation in the power supply, quality of the hardware (poor soldering or wiring), bits error in the data acquisition, etc, affect the quality of the measured output voltage force. A filter is therefore designed to improve the signal to noise ratio of the measured force signal.

![Figure C.1: Force sensor](image)

C.2 Filtering choice

Figure C.2 shows the unfiltered voltage force signal, captured with Dspace’s scope from a 0 volts measured output force sensor. The measured signal is mostly a noise signal present in the force sensor signal. The transient measurement shows a noise level with a magnitude of $-1.5\ volts$ to $1.5\ volts$ peak to peak, and the power spectrum density (PSD) shows the same signal, in its frequency domain within $500\ Hz$ bandwidth.
APPENDIX C. FORCE SENSOR’S FILTER DESIGN

Figure C.2: Unfiltered force signal
In this project, a digital filtering technique was chosen instead of an analog hardware filtering technique, upon the following criteria,

- **Thermal stability**: Temperature changes affecting resistors, capacitors, and inductors are eliminated.

- **Precision**: Filtering performance enhancement can be obtained, with respect to accuracy, dynamic range, stability, and frequency response tolerance, by increasing the processor register bit-length.

- **Adaptability**: Digital filter frequency response can efficiently be changed by reading a new set of filter coefficients. Unlikely in analog, a complete redesign of the hardware should always be made.

- **Cost**: no extra cost is added.

Nonetheless, this filtering technique has also some drawbacks which are

- **Limited bandwidth**: The bandwidth of the filter is limited to half a sampling frequency

- **Finite register length effects**: Higher the order of the filter, longer the length of the register will be. Therefore, this will have an impact on the microprocessor.

A compromised solution is needed between the order of the filter and the length of the register.
C.3 Denoising filter design

A lowpass filter using a direct-form realization of FIR with an Exponential Weighted Moving Averaging coefficients structure, illustrated in figure C.3 is designed to denoise the force signal.

The filtering structure is based on the finite impulse response (FIR) technique, which is known to be causal and has a linear phase delay. Obtaining a linear phase or a constant phase delay between the input \( x[n] \) and the output \( y[n] \), was an important matter for this project, due to the sudden change of the applied force from the user. The difference equation (David et al., 1988) of the filter structure of figure C.3 is derived as followed,

\[
y(iT) = \sum_{n=0}^{(N-3)/2} h(n) \{ x[(i-n)T] + x[(i+n+1-N)T] \} + h\left(\frac{N-1}{2}\right) x\left[(i-\frac{N-1}{2})T\right], \tag{C.1}
\]

where \( y(iT) \) denotes the output filter, \( x \) denotes the input, \( h(n) \) denotes the impulse function, \( T \) and \( N \) denote the time step and the total number of lags respectively. Similarly, the z-domain transfer function is also derived from equation (C.1) as,

\[
H(z) = \sum_{n=0}^{(N-3)/2} h(n) \left[ z^{-n} + z^{-(N-1-n)} \right] + h\left(\frac{N-1}{2}\right) z^{-(N-1)/2} \tag{C.2}
\]

The noise level is efficiently reduced by choosing for \( N = 51 \), knowing \( T = 0.002[s] \). But, these lags create a delay of,

\[
\tau_d = \frac{N-1}{2} T = \frac{50}{2} 0.002 = 0.05[s] \tag{C.3}
\]
Exponential smoothing assigns exponentially decreasing weights, as the observations get older. This technique is applied to determine the coefficients of each impulse response as followed,

$$\alpha = \frac{bT}{bT + T}, \quad \text{with} \quad b = \frac{N - 1}{2}$$

$$h(j) = \alpha^{j-1}(1 - \alpha),$$

and $$j = 1 \cdots b + 1$$

Figure C.4 shows graphically, the selection of the impulse’s coefficients using the Exponentially weighted moving averaging method. Figure C.5 shows the transient measurement and the power spectrum density of the same measurement as in figure C.2, after filtering. The peak to peak noise level is reduced by a factor of 20 according to the result obtained (figure C.5a), while the power spectrum density (figure C.5b) shows an attenuation of almost 20 dB.

![Figure C.4: Filter’s coefficients using EWMA](image-url)
APPENDIX C. FORCE SENSOR’S FILTER DESIGN

Figure C.5: filtered force signal
C.4 Filtering results

Figure C.6 shows the results of the filtering process. It is clearly illustrated on figure C.6 that, despite a delay of 0.05 sec, the designed filter eliminates quite well the noise from the force sensors signal.

![Unfiltered vs filtered force signal](image)

Figure C.6: Measured force signal
Appendix D

Simulation model overview

Figure D.1: Simulation model’s overview
Figure D.2: dynamic model
Figure D.3: Workspace control overview

Figure D.4: Workspace control overview 2
Figure D.5: Workspace control overview 3
Appendix E

Matlab files

E.1 Filter parameter’s file

%--------------------------------------------------------------------------
% 2Dof Haptic Gearshift 'C.Lever' Project @DTI-AM - 2009
% FIR filter with exponentially weighted moving average (FIR-EWMA)
% By A.Ayemlong Fokem: a.ayemlong.fokem@student.tue.nl
% Technische Universiteit Eindhoven - The Netherlands
%--------------------------------------------------------------------------

% Add the filter in simulink file in your program
% Run this script to load parameters in Matlab workspace
% Execute your program for simulation or offline filtering
% Build or compile the filter from simulink to C code for hardware execution
% This filter can also be implemented with other programming languages

% Parameters

Fs = 500; % sample frequency; frequency of your system
Ts = 1/Fs; % sample time
b = 25; % Number of delays
alpha = 0.95; % (b*Ts)/(b*Ts+Ts); % filtering weight,
alpha = 0.95; % Exponential weighted moving average algorithm
kappa = alpha; % " should not be larger than 1"

for i=1:b
    W(i)=kappa^(i-1);*(1-alpha);
end

E.2 Friction model estimation

function F1 = friction1(y,dtheta1)

Tc=y(1); Ts=y(2); dqs=y(3); etta=y(4); B=y(5); %Tz=x(7);delta=x(8);

S = (B*dtheta1+Tc+(Ts-Tc).*exp(-abs(dtheta1)).^(etta)).*tanh(1*dtheta1);
F1 = 0.9*S;
function F2 = friction(x,dtheta2)
  \%
  Tc=x(1); Ts=x(2); dqs=x(3); etta=x(4); B=x(5);
  \%
  S = (B*dtheta2+Tc+(Ts-Tc)*exp(-abs(dtheta2)./dqs).^etta).*tanh(1*dtheta2);
  F2 = 1*S;
end

clc;
y0 = [.2; 1.8033; 500; 20; 0.000050]; % Starting guess 1
x0 = [1.4; 1.2633; 150; 2.2; -0.00050]; % Starting guess

m=0.005;
lb = [];
ub = [];
  \%
for i=1:5
  options = optimset('TolX',1e-3,'TolFun',1e-3);
  [y,resnorm,residual,exitflag] = lsqcurvefit(@friction1,y0,dtheta1,Tf1,lb,ub,options);
  TF1 = m*Fa1
          +(y(5)*abs(dtheta1)+y(1)+(y(2)-y(1))*exp(-abs(dtheta1)./y(3)).^(y(4))).*tanh(1*dtheta1);
y1=y
end

for i=1:5
  options = optimset('TolX',1e-9,'TolFun',1e-3);
  [x,resnorm,residual,exitflag] = lsqcurvefit(@friction,x0,dtheta2,Tf2,lb,ub,options);
  TF2 = m*Fa2
          +(x(5)*abs(dtheta2)+x(1)+(x(2)-x(1))*exp(-abs(dtheta2)./x(3)).^(x(4))).*tanh(1*dtheta2);
x1=x
end
close all

figure;
plot(dtheta1(1,1:2000)*pi/180,Tf1(1,1:2000),'*','color',[0.4;0.4;0.4]);
    hold on
plot(dtheta1(1,1:2000)*pi/180,TF1(1,1:2000),'-','color',[0.8;0;0],'LineWidth',3)
axis([-1 1 -2 2])
hold off
xlabel('DD1');ylabel('Tf1');
title('TfD1');
legend(' Data'); \%,'Prediction',

APPENDIX E. MATLAB FILES
E.2. FRICTION MODEL ESTIMATION

yticks=[-2 -1 0 1 2];
xticks=[-1 -0.5 0.5 1];
set(gca,'ytickmode','manual','ytick',yticks,'yticklabel',{’yt0’,’yt2’,’yt3’,’yt4’,’yt5’});
set(gca,’xtickmode’,’manual’,’xtick’,xticks,’xticklabel’,{’xt0’,’xt2’,’xt3’,’xt4’})
grid

figure;
plot(dtheta2(1,1:2000)*pi/180,Tf2(1,1:2000),’*’,’color’,[0.4;0.4;0.4]);
hold on
plot(dtheta2(1,1:2000)*pi/180,TF2(1,1:2000),’-’,’color’,[0.8;0;0],’LineWidth’,3)
axis([-1 1 -2 2])
hold off
xlabel(’DD2’);ylabel(’Tf2’);title(’TfD2’);
legend(’ Data’); % ’Prediction’
yticks=[-2 -1 0 1 2];
xticks=[-1 -0.5 0.5 1];
set(gca,’ytickmode’,’manual’,’ytick’,yticks,’yticklabel’,{’yt0’,’yt2’,’yt3’,’yt4’,’yt5’});
set(gca,’xtickmode’,’manual’,’xtick’,xticks,’xticklabel’,{’xt0’,’xt2’,’xt3’,’xt4’})
grid