Friction in Robotic Manipulators

Modelling, Compensation and Simulation

Ola Jacob Mjøen Iversen
olajacob@pvv.ntnu.no

Trondheim, June 4th, 2002

Department of Engineering Cybernetics
Norwegian University of Science and Technology

NTNU
Norges teknisk-naturvitenskaplige universitet

Fakultet for informasjonsteknologi matematikk og elektronikk

Institutt for teknisk kybernetikk

HOVEDOPPGAVE

Kandidatens navn: Ola Jacob Mjøen Iversen

Fag: Teknisk Kybernetikk

Oppgavens tittel (engelsk): Friction in Robotic Manipulators: Modelling, Compensation and Simulation

Oppgavens tekst: This text is based on Iversens project work from the autumn 2001, [1].

1. Consider a dynamic model of a manipulator with dynamic friction. Compare the performance of control (tracking) systems for this system when the compensation scheme is based on a) Dynamic friction models (LuGre and elasto-plastic) and b) Static friction models. Are there benifits of using dynamic friction models in the compensator? Does the use of the elasto-plastic model eliminate the drift mentioned in [1]?

2. Assume that the parameters in the friction models are unknown and possibly timevarying. Develop adaptive versions of the tracking controller schemes in [1]. Establish the stability properties of the closed loop manipulator/observer/adaptive controller.

3. Study the possibility of using a neural network for in the friction compensator when the Stribeck curve of the friction model is unknown.

4. Compare experiments and simulations of a DC motor with friction. Use both static and dynamic friction models.

[1] Iversen, O.J.M. "Friction in Robot Manipulators: Modelling, Compensation and Simulation", Dept. of Engineering Cybernetics, 2001

Oppgaven gitt: 7/1-2002

Besvarelsen leveres: 4/6-2002

Besvarelsen levert: 4/6-2002

Utført ved Institutt for teknisk kybernetikk

Trondheim, den 7/1-2002

Jan Tommy Gravdahl

Preface

During one semester I have been working on my Master of Science thesis. I was supervised by associate professor Dr. Tommy Gravdahl at Department of Engineering Cybernetics. Industry partner ABB Corporate Research was represented by Dr. Dag Kristiansen.

The technical field of modelling and control of nonlinear and uncertain systems are really interesting. I have learned a lot which I hope to benefit from in later occations.

I want to thank my advisor Dr. Tommy Gravdahl for the motivation and help during this work. Secondly I want to thank my office mate Joar Holmefjord for providing me with experimental data from ABB Billingstad.

Trondheim june 4th 2002

Ola Jacob Mjøen Iversen

Abstract

This thesis is focusing on friction phenomena in robotic manipulators. An introduction to these phenomena presented the two main categories of friction models, *static* and *dynamic* friction models. *Dynamic* friction models behave more like real friction, while *static* friction models don't include important friction phenomena and thus have poor performance at low velocities and when crossing zero velocity. This was shown by experimental data of a DC motor and comparing simulation studies.

The dynamic LuGre model has been verified to exhibit drift, however the drift is more noticeable when the control signal is exposed to noise. Moreover it was shown that the numerical value of this drift over the elapsed time, was quite small.

An adaptive compensation scheme was developed to estimate three unknown parameters in the LuGre friction model in connection to a N-DOF robotic manipulator. It was assumed that the Stribeck phenomenon was known. The adaptive controller depends upon a unknown internal friction state which was estimated through a dual-observer. The closed-loop and the position tracking error was shown to be globally asymptotically stable. Nevertheless, the shown bounded parameters didn't converge to their true values, since this was not proven in the stability analysis.

Finally a study of a different approach using neural networks to find the unknown Stribeck phenomenon were performed. It was concluded that it is not straight forward to apply this scheme to the Elasto-Plastic friction model.

Contents

List of Figures

Chapter 1

Friction in Robot Manipulators

To boldly go where no map has gone before

1.1 Introduction

Mechanical systems such as motors, manipulators etc. includes dynamics that are difficult to model exact. These phenomena are undesired in control systems and needs to be canceled out in order to obtain satisfactorily results. Researchers have been studying friction phenomena for decades and have discovered the different friction phenomena by experimental studies. As a result there have been established several mathematical formulas to describe the different phenomena. Friction can be divided into two categories: *static* and *dynamic* models. Static models are straight forward mathematical expressions that are easy to implement and are widely used in the industry. Dynamic models includes more phenomena and are more exact. At low velocities, friction becomes considerably significant and can cause severe errors in the control system if not handeled properly.

One important obstacle in friction compensation is the parameter uncertainties. Friction forces varies with time, temperature and metal properties. Off-line estimates of the different parameter gives rise to increased uncertainties over time and requires frequent system calibrations. On-line parameter estimating increases the flexibility and gives more reliable results. In this work we will first assume that the friction parameters are constant and known. Secondly we will treat the case when the parameters are unknown.

1.2 Motivation

Since the industry are using mostly static friction models, we are motivated to investigate if there are any benefits of using dynamic friction models in the control system. By

considering a dynamic model of a manipulator with dynamic friction we will be comparing the performance of control (tracking) systems when the compensation scheme is based on dynamic and static friction models. This will be under the assumption that the parameters are known. Secondly we will assume the friction parameters being unknown and develop an adaptive controller. There are two approaches to cope with uncertainties: 1) Live with it and use robust control, or 2) reduce it and use adaptive control.

An introduction to established friction theory should provide the reader with insight into the different friction phenomena. This survey is given in chapter 2 and is focusing on the friction terminology and mathematical models from an control engineering point of view.

1.3 Organization of the Thesis

This report is organized into 7 chapters and an appendix

- Chapter 2 gives a survey of friction modelling and comparison of some of them.
- Chapter 3 presents modelling of experimental setup, the manipulator, inverse kinematics and controller design.
- Chapter 4 is divided into 4 main sections, where we present different compensation schemes that will be simulated.
- Chapter 5 provides the experimental results and simulation results.
- In chapter 6 we discuss the results of this thesis
- Chapter 7 contain the main results and conclusions, and proposes further work
- The appendix provides tools for analysis and lists the main source code and Simulink diagrams.

Chapter 2

Friction - A theoretical foundation

In this chapter, a brief introduction to friction phenomena, both static and dynamic models, is presented. Friction has been subject for active research in several fields. [2] gives a survey of models, analysis tools and compensation methods for the control of machines with friction. For a more thorough introduction on friction phenomena, please refer to [1] and [19]. A recent article discussing the elasto-plastic model is [6]. Interested readers should also study [18].

2.1 Friction Phenomena

Friction is the tangential reaction force between two surfaces in contact, Figure 2.1. This force is proportional to the normal force as the well known classical friction force $F = \mu N$. The apparent contact of the surfaces is not the true area of contact. Asperities between the surfaces makes the friction force to depend up on the properties of the materials, displacements, relative velocity and lubrication.

Figure 2.2 demonstrates a model of the contact between two surfaces in a microscopic scale. The surfaces are constructed by small asperities. The true contact occurs between these asperities, in what are called asperity junctions. In engineering materials the slopes of the asperities are typically 5 to 10 degrees, whereas the junction widths typically are $1 \cdot 10^{-5}$ (in steel). The true area of contact is therefore much smaller than the apparent area of contact and is determined by

$$\text{true area of contact} = \frac{\text{load}}{\text{yield pressure}},$$

where "yield pressure" is a material property. The asperities deform to generate the contact area necessary to take up the total load.

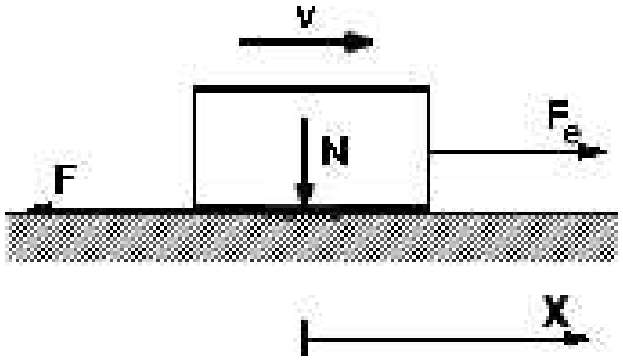


Figure 2.1: Friction force

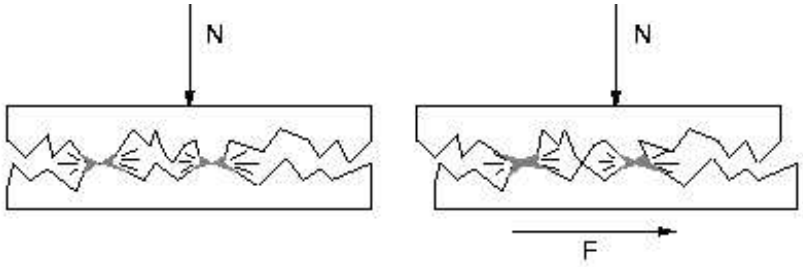


Figure 2.2: Contact between asperities

Experiments on some surfaces have indicated that friction is less when surfaces are dry, [19]. By dry friction one mean absence of pollution between the surfaces. This is only an ideal situation that are not possible to achieve in real mechanical installations because there will always be a chemical connection between materials in contact.

Descriptions of different friction phenomena is given in the literature. These have been attempted identified. Some of them are well explained, while others are less known. When the velocity isn't constant, the dynamics of the model is becoming important and gives rise to different phenomena.

Static friction (stiction) The force (torque) needed to put a body into movement.

Stic-slip movement Stic-slip movement is caused by the fact that the force is greater when the body is at rest than when in movement. This effect will not appear when the force is big enough, as can be seen in Figure 2.3 b). The setup is modelled as a box being pulled by a force F connected in series with a spring with a constant k, see Figure 2.3 a). The experiment was carried out by Canudas de Wit in [5]. In control systems, the stic-slip is highly undesirable and a good model is desired in order to achieve a precise

compensating.

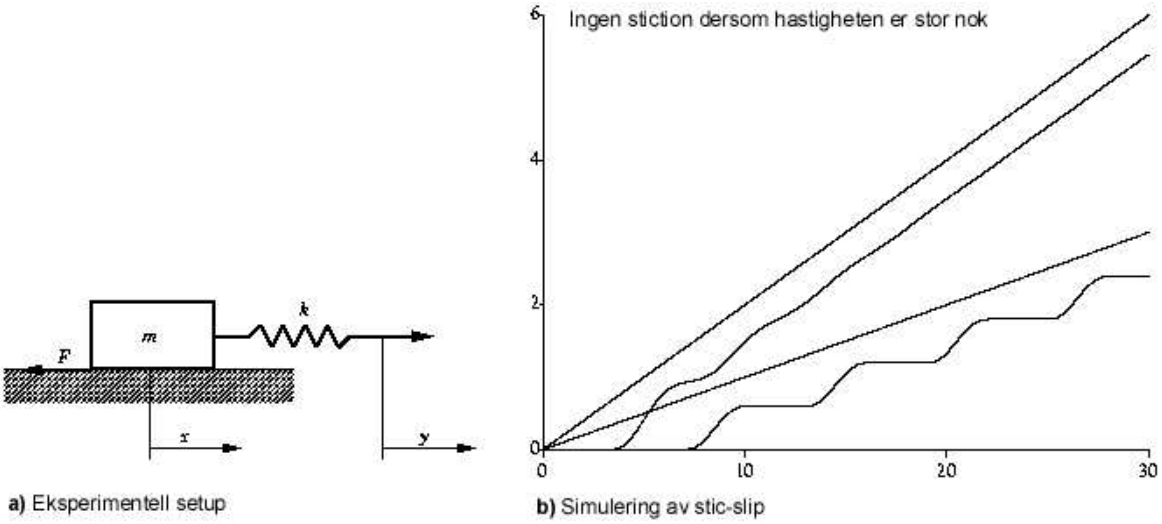


Figure 2.3: Stic-slip movement

Dynamic (Coulomb) friction A friction component independent of the velocity.

Frictional lag Frictional lag, see hysteresis.

Viscous friction Velocity dependent friction between body and lubricate.

Pre-sliding displacement Small movements in the elastic area are called pre-sliding displacement or the Dahl-effect. The area where the sliding is elastic is normally less than 2 micro for steel. This behaviour can be described as a spring when the applied force is less than the force putting the body into sliding (break-away).

Stribeck effect At low velocities the friction force will decrease at increased velocity. This is referred to as the Stribeck-velocity.

Hysteresis Hysteresis appears in the relation friction - velocity. Hess and Soom showed the friction force is less at decreasing velocity than at increasing velocity.

Varying "break-away" forces Break-away is the transition from stiction into sliding (dynamic friction). The amount of force needed to break the static friction is called the break-away force.

Dissipative properties Friction is dissipative, which imply that it consumes energy from a system. This is a very desired property of a friction model in control systems.

2.2 Static models

The simplest models describes the friction as a function of the relative velocity between the gliding surfaces. These models are often referred to as static models. There are several problems with these models, and one of them appears when the velocity crosses zero, $v = 0$. Then the friction isn't a function of velocity, which implies mathematical complexities. Simulation of such models produces practical difficulties as a result of the problem of telling when the velocity precisely reaches zero.

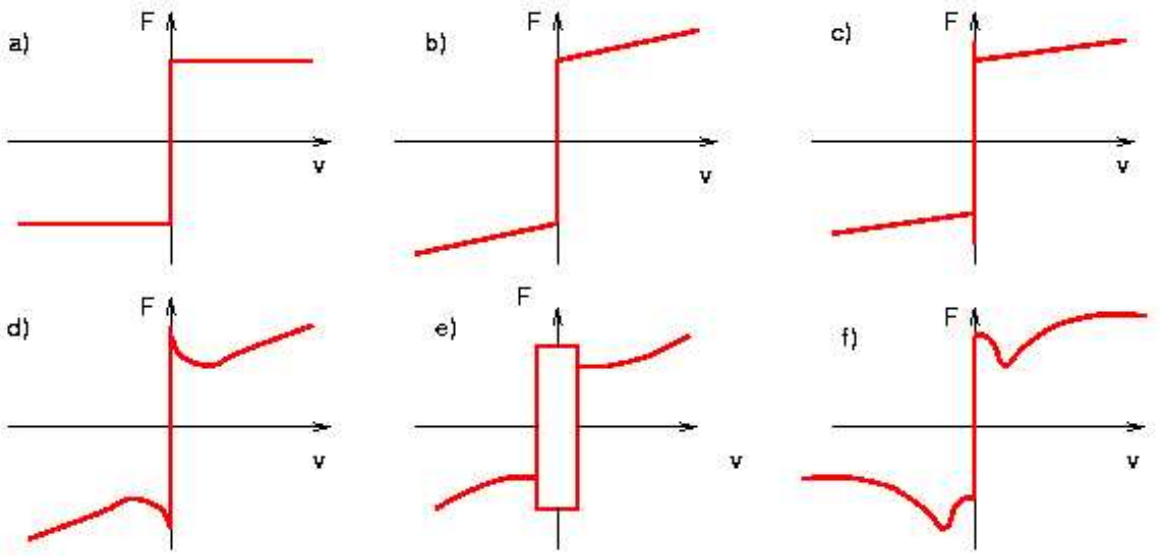


Figure 2.4: a) Coulomb b) Coulomb + viscous c) Coulomb + viscous + stiction d) Stribeck effect e) Karnopp f) Hess and Soom

2.2.1 Coulomb-friction

The Coulomb-friction depict the relation between friction and force. This model is simple and linear, see Figure 2.4 a).

$$F = F_C sgn(v) = \mu F_N sgn(v), \tag{2.1}$$

where F_C is the friction force, F_N is normal load and μ is the friction coefficient. This model was first discovered by Leonardo da Vinci (1591), but was rediscovered by Amonts (1699) and developed by Coulomb (1785), [2].

An extension and improvement of the Coulomb-friction was later done by including viscosity as a proportional friction force with respect to velocity $F = F_R v$, Reynolds (1886). This extension follows by the viscosity of the lubricants. Figure 2.4 b) shows Coulomb with viscous extension. Experiments has shown that the parameters F_C and F_R often depends on the direction of the velocity.

The model given by equation (2.1) does not describe the phenomenon when zero-velocity because the signum function isn't defined when $v = 0$. Applying a small force is causing the friction to behave like a spring; and this is the phenomenon called "stiction" or static friction. This is an extra friction force at zero velocity which can be seen graphically in Figure 2.4 c). By extending the Coulomb-model, this model appears

$$F = \begin{cases} F_e, & \text{if } v = 0 \text{ and } |F_e| < F_S; \\ F_S sgn(F_e), & \text{if } v \neq 0 \text{ and } |F_e| \geq F_S. \end{cases} \tag{2.2}$$

where F_e is the force applied on the body. The model can be looked at as a composition of two models, one when $v = 0$ and a Coulomb friction model when $v \neq 0$. The model when $v = 0$ implies that the friction has opposite direction relative to the velocity as long as applied force is less than the stiction force F_S.

2.2.2 Stribeck-friction

In 1902 professor Stribeck discovered -by experiment- that the friction force does not suddenly drop as illustrated in Figure 2.4 c). Stribeck [18] observed a friction force with a falling shape as illustrated in Figure 2.4 d). The velocity v_s where the friction force is at minimum is named the *Stribeck-velocity*. The Stribeck model is described as

$$F = \begin{cases} F(v), & \text{if } v \neq 0; \\ F_e, & \text{if } v = 0 \text{ and } |F_e| < F_S; \\ F_S sgn(F_e), & \text{else} \end{cases} \tag{2.3}$$

The function $F(v)$ can be identified by measuring the force that is needed to maintain constant velocity. Mostly it is asymmetric, i.e. direction dependent.

Armstrong-Hélvoury (1990) proposed the following expression of $F(v)$:

$$F(v) = F_C + (F_S - F_C)e^{-|v/v_s|^{\delta_s}} + F_R v \tag{2.4}$$

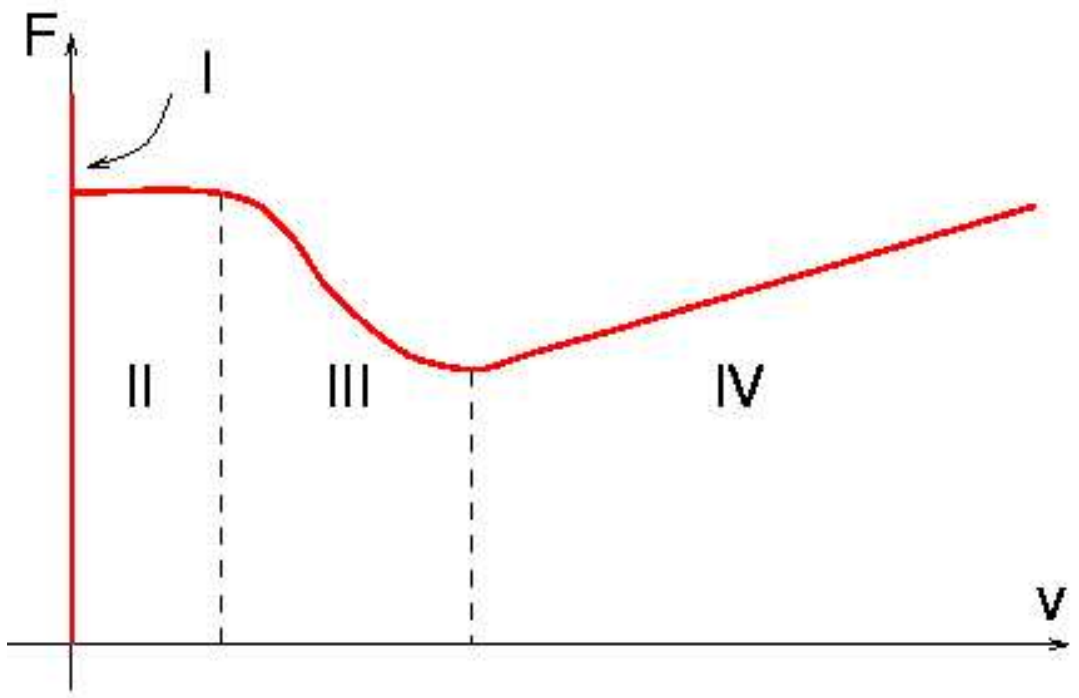


Figure 2.5: The generalized Stribeck curve

The Stribeck effect shall produce a destabiliating effect at low velocities.

I No sliding, elastic deformation.

II Boundary lubrication

III Partial fluid lubrication

IV Full fluid lubrication

The first regime: static friction and presliding displacement. Contact is shown to occur at asperity junctions. From the standpoint of control, these junctions have two important behaviours: they deform elastically, giving rise to presliding displacement; and both the boundary film and the asperities deform plastically, giving rise to static friction.

The second regime: boundary lubrication. In the second regime - that of very low velocity sliding - fluid lubrication is not important, the velocity is not adequate to build a fluid film between the surfaces.

The third regime: partial fluid lubrication. Imagine a process where lubricant is drawn into the contact zone as Figure 2.6. Lubricant is brought into the load bearing region through motion, either by sliding or rolling. Some is expelled by pressure arising from the load, but viscosity prevents all of the lubricant from escaping and thus a film is formed. The entrainment process is dominated by the interaction of lubricant viscosity, motion speed and contact geometry. The greater viscosity or motion velocity, the thicker the fluid film will be. When the film is not thicker than the height of the asperities, some solid-to-solid contact will result and there will be partial fluid lubrication. When the film is sufficiently thick, separation is complete and the load is fully supported by fluid. Partial fluid lubrication is the most difficult to model of the four regimes, [2].

The fourth regime: full fluid lubrication. Hydrodynamic or elasto-hydrodynamic. Hydrodynamic and elasto-hydrodynamic lubrication (EHL) are two forms of full fluid lubrication. Hydrodynamic lubrication arises in conformal contacts, and EHL in non conformal contacts. Solid-to-solid contact is eliminated. In this regime, wear is reduced by orders of magnitude and friction is well behaved. The object of lubrication engineering is often to maintain full fluid lubrication effectively and at low cost.

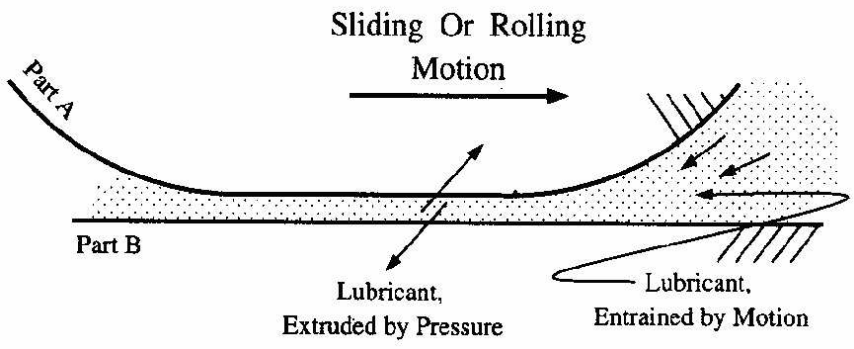


Figure 2.6: Motion brings fluid lubricant into the contact zones.

2.2.3 Problems with the static models

Two main problems regarding static friction models were studied by Karnopp and Armstrong. Nevertheless, static friction models lack significant properties which becomes critical at low velocities.

1. They are dependent on detection of zero velocity, since switching between two different equations are done
2. They do not describe all observed dynamical effects, such as pre sliding displacement, varying break away force and frictional lag.

Karnopp model

In 1985 Karnopp presented a model as illustrated in Figure 2.4 e). This model includes a dead-zone at zero velocity. The models described above has a disadvantage when identifying zero velocity. In Karnopps model the condition $v = 0$ is replaced by $|v| < \epsilon$. The model is an improvement of the stiction models, but introduces difficulties because the behaviours depends strongly by the choice of ϵ and the robustness of numerical integration routines. An another drawback with the model is that it is strongly coupled to the rest of the system. The external force is an input to the model and this force is not always given explicitly. One can also observe that the models are discontinuous at $v = 0$. Karnopp showed that an approximation with finite slope through the origin wouldn't reflect the physical phenomenon.

Armstrongs model

A classical model has been modified by [2] to account for some of the observed dynamics in friction phenomena. This model introduces time dependency for stiction and the Stribeck effect, but does not handle pre-sliding displacement. This was handeled by describing the sticky behaviour with a separate equation. Then a mechanism must control the recoupling between the stiction model and the sliding model. The friction is described by

$$F(x) = \sigma_0 x \tag{2.5}$$

when sticking and by

$$F(v,t) = \left(F_C + F_S(\gamma, t_d) \frac{1}{1 + (v(t-\tau_l)/v_s)^2} \right) sgn(v) + F_v v \tag{2.6}$$

when sliding, where

$$F_S(\gamma, t_d) = F_{S,a} + \left(F_{S,\infty} - F_{S,a} \frac{t_d}{t_d + \gamma} \right) \tag{2.7}$$

$F_{S,a}$ is the Stribeck friction at the end of previous sliding period and t_d the dwell time, i.e., the time since becoming stuck.

2.3 Dynamic Models

Studies has shown that friction models should include dynamics in order to describe the friction phenomena more precise. The desire to obtain better precision and the use of

better hardware makes it possible to implement friction compensators.

2.3.1 Dahl Model

The Dahl model is inspired by the stress-strain characteristics from solid mechanics. As a consequence the friction force is described as only dependent of the position. The model offers a good description of the unidimentional phenomenon when the transient behaviour is elastic rather than viscous. When subject to stress the friction force increases gradually until rupture occurs. Dahl modeled the stress-strain curve by a differential equation. Let x be the displacement, F the friction force, and F_c the Coulomb friction force. The Dahl's model has the form

$$\frac{dF}{dx} = \sigma \left(1 - \frac{F}{F_c} sgn(v)\right)^{\alpha} \tag{2.8}$$

where σ is the stiffness coefficient and α is a parameter that determines the shape of the stress-strain curve. The value $\alpha = 1$ is most commonly used. Higher values will give a stress strain curve with a sharper bend. The friction force $|F|$ will never be larger than F_c if its initial value is such that $|F(0)| < F_c$ [19].

To obtain the time domain model Dahl observed that

$$\frac{dF}{dt} = \frac{dF}{dx}\frac{dx}{dt} = \frac{dF}{dx}v = \sigma \left(1 - \frac{F}{F_c} sgn(v)\right)^{\alpha} v \tag{2.9}$$

For $\alpha = 1$ the Dahl model (2.9) reduces to

$$\frac{dF}{dt} = \sigma v - \frac{F}{F}|v|. \tag{2.10}$$

Introducing $F = \sigma z$ the model can be written as

$$\frac{dz}{dt} = v - \frac{\sigma|v|}{F_c} z, \tag{2.11}$$

$$F = \sigma z \tag{2.12}$$

The Dahl model offers a regularization of Coulomb friction at velocity zero crossings, but does not include stiction nor captures the Stribeck effect.

2.3.2 Bristle Model

Haessig and Friedland introduced this friction model, which attempted to capture the behaviour of the microscopical contact points between two surfaces. Due to irregularities in the surface the number of contact points and their location are random. Each point of contact is thought of as a bond between flexible bristles. As the surfaces move relative to each other the strain in the bond increases and the bristles act as springs giving rise to friction force. The force is given by

$$F = \sum_{i=1}^{N} \sigma_0 (x_i - b_i) \tag{2.13}$$

where N is the number of bristles, σ_0 the stiffness of the bristles, x_i the relative position of the bristles, and b_i the location where bond was formed. As $|x_i - b_i|$ equals δ_s the bond snaps and new one is formed at a random location relative to the previous location.

The complexity of the model increases with N. Good results where found with 20-25 bristles, but even a single bristle gave reasonable qualitative behaviour, [19]. The stiffness of the bristles, σ_0, can be made velocity dependent. An interesting property of the model is that it captures the random nature of friction. The randomness depends on the number of bristles. The model is inefficient in simulations due to its complexity. Motion in sticking may be oscillatory since there is no damping of the bristles in the model.

2.3.3 Bliman-Sorine model

Bliman and Sorine stress rate independence. The magnitude of the friction depends only on the $sgn(v)$ and the space variable s defined by

$$s = \int_0^t |v(\tau)| d\tau. \tag{2.14}$$

In the Bliman-Sorine models, friction is then a function of the path only. It does not depend on how fast the system moves along the path. This makes it possible to use elegant theory of hysteresis operators developed in [12, 3]. The models are expressed as linear systems in the space variable s

$$\frac{dx_s}{ds} = Ax_s + Bv_s \tag{2.15}$$

$$F = Cx_s \tag{2.16}$$

The variable $v_s = sgn(v)$ is required to obtain the correct sign. Bliman and Sorine have models of different complexity. The first order model is given by

$$A = -\frac{1}{\epsilon_f}, \quad B = \frac{f_1}{\epsilon_f} \quad \text{and} \quad C = 1. \tag{2.17}$$

This model can be written as

$$\frac{dF}{dt} = \frac{dF}{ds}\frac{ds}{dt} = |v|\frac{dF}{ds} = \frac{f_1}{\epsilon_f}\left(v - |v|\frac{F}{f_1}\right) \tag{2.18}$$

which is identical to the Dahl model with $F_C = f_1$, $\sigma = \frac{f_1}{\epsilon_1}$ and $\alpha = 1$. The first order model does not give stiction, nor does it give a friction peak at a specific break-away distance as observed by Rabinowicz. This can, however, be achieved by a second order model with

$$A = \begin{bmatrix} -\frac{1}{\eta\epsilon_f} & 0 \\ 0 & -\frac{1}{\epsilon_f} \end{bmatrix} \tag{2.19}$$

$$B = \begin{bmatrix} \frac{f_1}{\eta\epsilon_f} \\ -\frac{f_2}{\epsilon_f} \end{bmatrix} \tag{2.20}$$

where $f_1 - f_2$ corresponds to kinetic friction reached exponentially as $s \to \infty$. The model can be viewed as a parallel connection of a fast and a slow Dahl model. The fast model has higher steady state friction than the slow model. The force from the slow model is subtracted from the fast model, which results in a stiction peak. Both the first and the second order models can be shown to be dissipative. Bliman and Sorine also show that, as ϵ_f goes to zero, the first order model behaves as a classical Coulomb friction model, and the second order model as a classical model with Coulomb friction and stiction. It should be noted that the Stribeck effect of the second order model is not the same as observed in [18]. The emulated effect by the second order model is only present at a certain distance after motion starts. This means that it will not appear when motion slows down, as the true Stribeck effect would. The friction peak is instead the equivalent of stiction for a dynamic model.

2.3.4 LuGre Model

The LuGre model is another generalization of Dahl's model. The model is related to the bristle interpretation of friction. Friction is modeled as the average deflection force of elastic springs. When a tangential force is applied the bristles will deflect like springs. If the deflection is sufficiently large the bristles starts to slip. The average bristle deflection for steady state motion is determined by the velocity. It is lower at low velocities, which implies that steady state deflection decreases with increasing velocity. This models the phenomenon that the surfaces are pushed apart by the lubricant, and models the Stribeck effect. The model also includes rate dependent friction phenomena such as varying break-away force and frictional lag. The model has the form

$$\frac{dz}{dt} = v - \sigma_0 \frac{|v|}{g(v)} z, \tag{2.21}$$

$$F = \sigma_0 z + \sigma_1(v)\frac{dz}{dt} + f(v) \tag{2.22}$$

where z denotes the average bristle deflection. The model behaves like a spring for small displacements. Linearization around zero velocity and zero state gives

$$\frac{d(\delta z)}{dt} = \delta v, \tag{2.23}$$

$$\delta F = \sigma_0 \delta z + (\sigma_1(0) + f'(0))\delta v. \tag{2.24}$$

The parameter σ_0 is the stiffness of the bristles, and $\sigma_1(v)$ the damping. For constant velocity the steady state friction force is

$$F = g(v)sgn(v) + f(v) \tag{2.25}$$

The function $g(v)$ models the Stribeck effect, and $f(v)$ is the viscous friction. A reasonable choice of $g(v)$ which gives a good approximation of the Stribeck effect is

$$g(v) = \alpha_0 + \alpha_1 e^{-(v/v_s)^2} \tag{2.26}$$

The sum $\alpha_0 + \alpha_1$ then corresponds to stiction force and α_0 to Coulomb friction force. The parameter v_s determines how $g(v)$ vary within its bounds $\alpha_0 < g(v) \leq \alpha_0 + \alpha_1$. A common choice of $f(v)$ is linear viscous friction $f(v) = \alpha_2 v$ as in (2.12).

The following special case of the model given by equation (2.21), (2.22) and equation (2.26), which has linear viscous friction and constant σ_1, is called the *standard parameterization.*

$$\frac{dz}{dt} = v - \sigma_0 \frac{|v|}{g(v)} z \tag{2.27}$$

$$g(v) = \alpha_0 + \alpha_1 e^{-(v/v_s)^2} \tag{2.28}$$

$$F = \sigma_0 z + \sigma_1 \dot{z} + \alpha_2 v \tag{2.29}$$

It is useful to let the damping σ_1 decrease with increasing velocity, e.g.

$$\sigma_1(v) = \sigma_1 e^{-(v/v_d)^2} \tag{2.30}$$

Physically this is motivated by the change of the damping characteristics as velocity increases, due to more lubricant being forced into the interface. Another reason for using (2.30) is that it gives a model which is dissipative, with the mapping $v \mapsto \sigma_0 z + \sigma_1 \dot{z}$, which is the case when

$$\sigma_1 < 4 \frac{g(v)}{|v|} \tag{2.31}$$

The proof was shown in [10].

2.3.5 Elasto-Plastic Model

The Elasto-Plastic friction model is a new class of single state models in which presliding is elasto-plastic: under loading, frictional displacement is first purely elastic and then transitions to plastic.

The rigid body displacement x is decomposed into elastic and plastic components z and w:

$$x = z + w \tag{2.32}$$

Friction models typically define the elastic dynamics explicitly while the plastic displacement w is defined implicitly.

The elasto-plastic model is defined by the following equations

$$\dot{z} = \dot{x}\left(1 - \alpha(z,\dot{x})\frac{\sigma_0}{f_{ss}(\dot{x})}sgn(\dot{x})z\right)^i, \quad \frac{\sigma_0}{f_{ss}(\dot{x})} > 0, \quad i \in \mathbb{Z} \tag{2.33}$$

$$f_f = \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 \dot{x}, \quad \sigma_i > 0 \tag{2.34}$$

where the integer exponent i was originally used by Dahl to govern the transition rate of z in order to achieve a better experimental match. Applications of his model typically employ the value $i = 1$. The piecewise continuous function $\alpha(z,\dot{x})$ is defined as follows

$$\alpha(z,\dot{x}) = \begin{cases} 0, & |z| \le z_{ba}, \\ 0 < \alpha < 1, & z_{ba} < |x| < z_{max}(\dot{x}), \\ 1, & |z| \ge z_{max}(\dot{x}) \end{cases} \tag{2.35}$$

$$0 < z_{ba} < z_{max}(\dot{x}) = \frac{f_{ss}(\dot{x})}{\sigma_0}, \quad \forall \dot{x} \in \mathbb{R} \tag{2.36}$$

A breakaway displacement $z_{ba} > 0$ is defined such that the models behave elastically for $|z| < z_{ba}$. The model above possesses the following properties:

1. The state z is bounded: If $|z(0)| \le z_{max} = f_{max}/\sigma_0$ then $|z(t)| \le z_{max}, \forall t \ge 0$.
2. Super relaxation is precluded: $0 \le dz/dx \le 1$ if no Stribeck effect is modeled and $-\infty < dz/dx \le 1$ otherwise.
3. The model possesses a stiction phase: A breakaway displacement $z_{ba} > 0$ exists such that the model behaves elastically for $|z| < z_{ba}$.
4. During sliding, the friction force opposes slip: $sgn(f_f) = sgn(\dot{w}), \forall \dot{w} \neq 0$.
5. The model is dissipative for all $\dot{x} \neq 0$.

Property 5 states the dissipative property of the model. In [6] they considered the map $\dot{x} \mapsto f_f$, from rigid body velocity to friction force. Using $V(t) = 1/2\sigma_0 z^2$ as a positive definite storage function, we write the input-output product as

$$\dot{x} \mapsto f_f = (\dot{z} + \dot{w})(\sigma_0 z + \sigma_1 \dot{z}) + \sigma_2 \dot{x}^2 \tag{2.37}$$

$$= \dot{V} + \sigma_1 \dot{z}^2 + \dot{w} f_f + \sigma_2 \dot{x}^2 \tag{2.38}$$

$$> \dot{V}, \quad \forall \dot{x} \neq 0. \tag{2.39}$$

Which can be used to conclude that the map $\dot{x} \mapsto f_f$ is dissipative for any nonzero input. For simulation studies, $\alpha(z, \dot{x})$ is chosen to produce stiction and is given by

$$\alpha(z, \dot{x}) = \begin{cases} 0, & |z| \leq z_{ba}, \\ \frac{1}{2}\sin(\pi \frac{z - \left(\frac{z_{max} + z_{ba}}{2}\right)}{z_{max} - z_{ba}}) + \frac{1}{2}, & z_{ba} < |x| < z_{max}(\dot{x}), \\ 1, & |z| \geq z_{max}(\dot{x}) \end{cases} \tag{2.40}$$

and can be visualized as in Figure 2.7

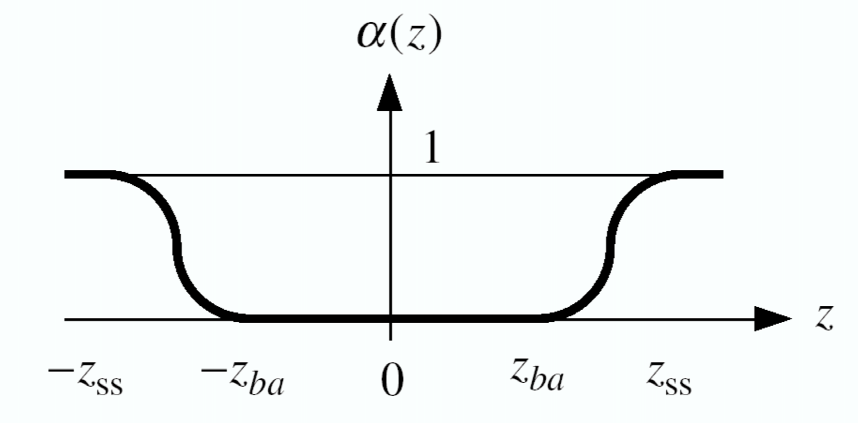


Figure 2.7: Plot of $\alpha(z, \dot{x})$ for $sgn(\dot{x}) = sgn(z)$

2.4 Comparisons of friction models

A comparison of the Bliman-Sorine and the LuGre models were investigated in [11]. Comparisons were made of captured friction phenomena, behaviour at zero crossings of the velocity and computational issues. It was concluded that the LuGre model exhibits a richer behaviour in terms of friction phenomena. The Bliman-Sorine model can be problematic to use because of poor damping properties at zero crossings of the velocity.

A recent article on dynamic models, [6], proposes the Elasto-Plastic Friction Model. The article showed that modeling presliding displacement and sliding in a single function gave rise to subtle issues. A comparison of phenomena in four different friction models pointed out that the stiction property of the LuGre model is not rendered as expected in earlier articles on the LuGre model. Table 2.1 shows the results of the comparisons.

Table 2.1: Comparison of four models of friction

Friction Model	**Stiction**	**Presliding Displacement**
Coulomb and Karnopp Models	Yes	No
Regularized Coulomb	No	Yes
LuGre Model	No	Yes
Elasto-Plastic Model	Yes	Yes

Models in common use possess several combinations of presliding displacement and stiction. the classic Coulomb friction mode, like Karnopp model, renders stiction but makes no reference to presliding displacement. Because of challenges posed by the discontinuity at zero velocity, the Coulomb friction model is sometimes regularized, leading to a model having neither presliding displacement nor stiction. The LuGre model renders presliding displacement but not stiction; and the elasto-plastic model renders both presliding displacement and stiction, [6]. The LuGre model gives rise to a steady drift in position during applied force. This is seen in Figure 2.8 from [7].

A similar simulation as in Dupont is visualised in Figure (2.9). The oscillatory applied force represents noise.

As seen from Figure 2.10 the drift is not that significant without "noise" as when noise is introduced to the system, Figure 2.9. The force applied to the LuGre system in Figure 2.10 is similar to the force in 2.9 except for the oscillatory movement.

2.5 Friction in Servomotors

In a manipulator control system the friction arise in the joint actuation system, [16]. The joint motion is entrusted to *motors* which allow realizing a desired motion for the mechanical system. The motors employed in robotics are the evolution of the motors employed in industrial automation having powers ranging from about ten watts to about ten kilowatts. For the typical performance required, such motors shall have the following requirements with respect to those employed in conventional applications:

- low inertia and high power-to-weight ratio,
- possibility of overload and delivery of impulse torques,
- capacity to develop high accelerations,
- wide velocity range (from 1 to 1000),
- high positioning accuracy (at least 1/1000 of a circle),
- low torque ripple so as to guarantee continuous rotation even at low speed

These requirements are enhanced by the good trajectory tracking and positioning accuracy demanded to an actuating system for robots, and thus the motor shall play the role of a *servomotor*.

To control an *electric servomotor*, it is necessary to provide it with a voltage or current of suitable form depending on the kind of servomotor employed. Voltage (or current) is direct for permanent-magnet dc servomotors, while it is alternating for brushless dc servomotors. The value of voltage for permanent-magnet dc servomotors or the values of voltage and frequency for brushless dc servomotors are determined by the control signal of the amplifier, so as to make the motor execute the desired motion, [16].

By customising the electric servomotors to the specific application in robotic control, one will contribute to reduce the control difficulties. From a control engineers point of view, a basic study of the used actuator system should point out the control scheme in practical applications.

A common way of representing friction in servomotors is by adding viscous friction $B\omega$ as shown in [8]

$$J\dot{\omega} = \tau - B\omega - T_L \tag{2.41}$$

where T_L is the load.

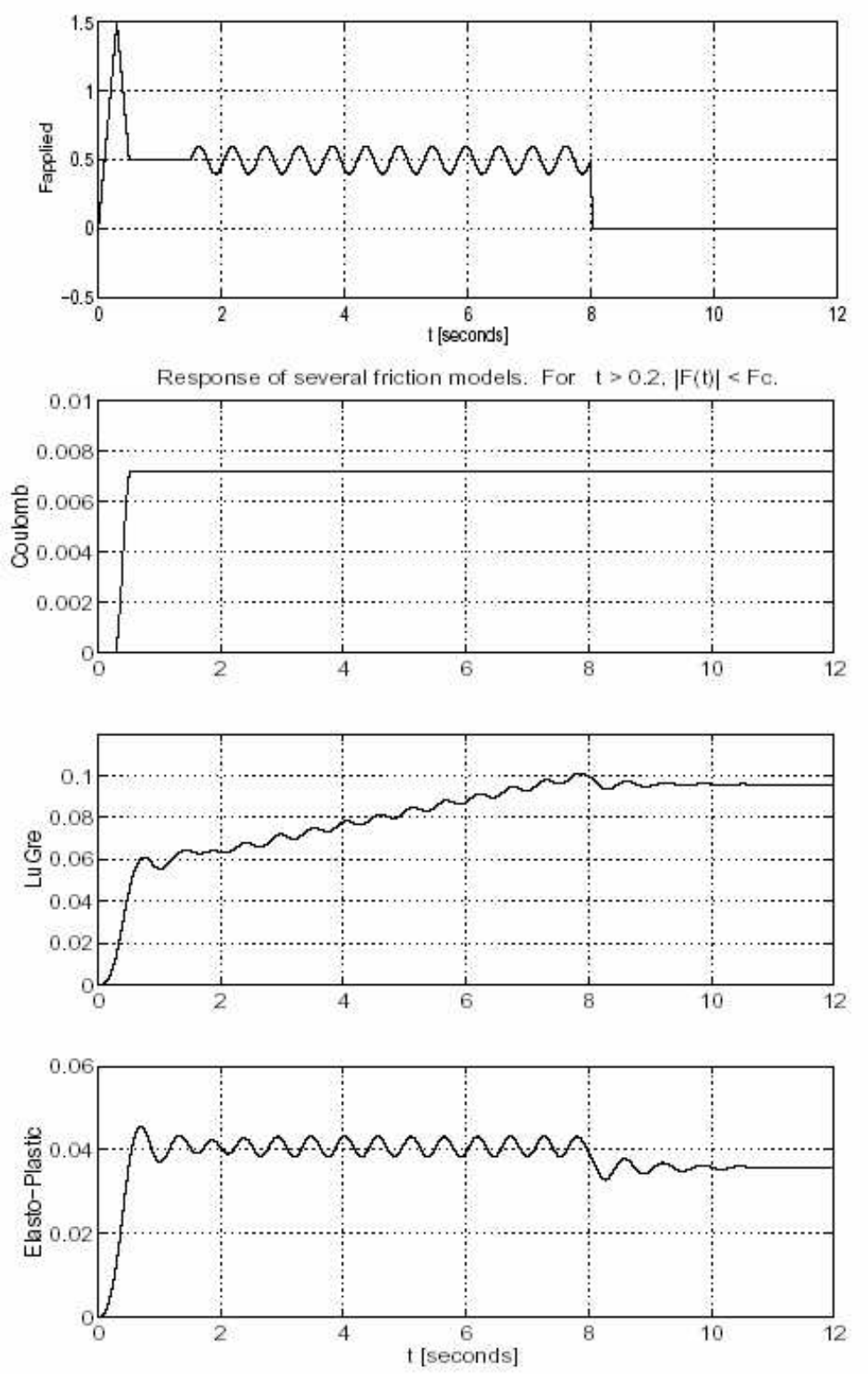


Figure 2.8: Top panel: force applied to a block in frictional contact with planar surface. Other panels: resulting displacement of a block using three friction models.

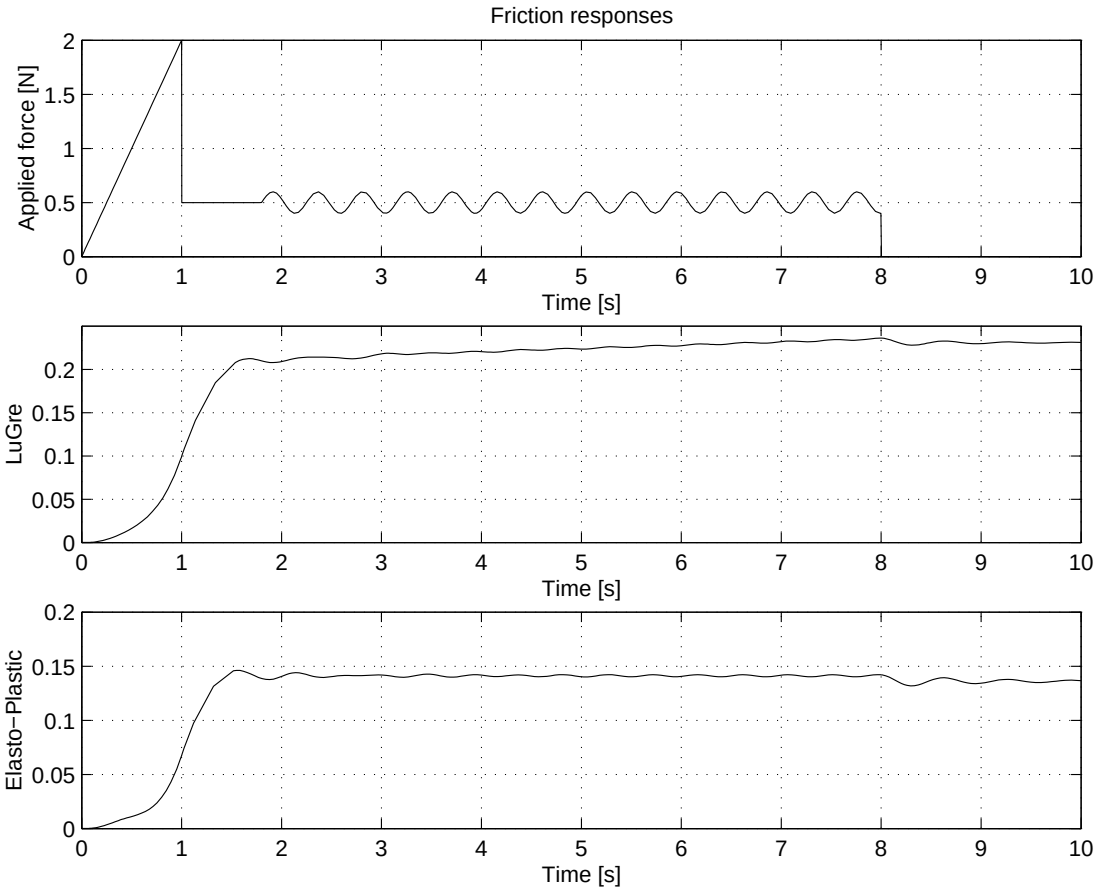


Figure 2.9: Similar simulation as in Dupont. Responses is given in displacement x(t) of mass

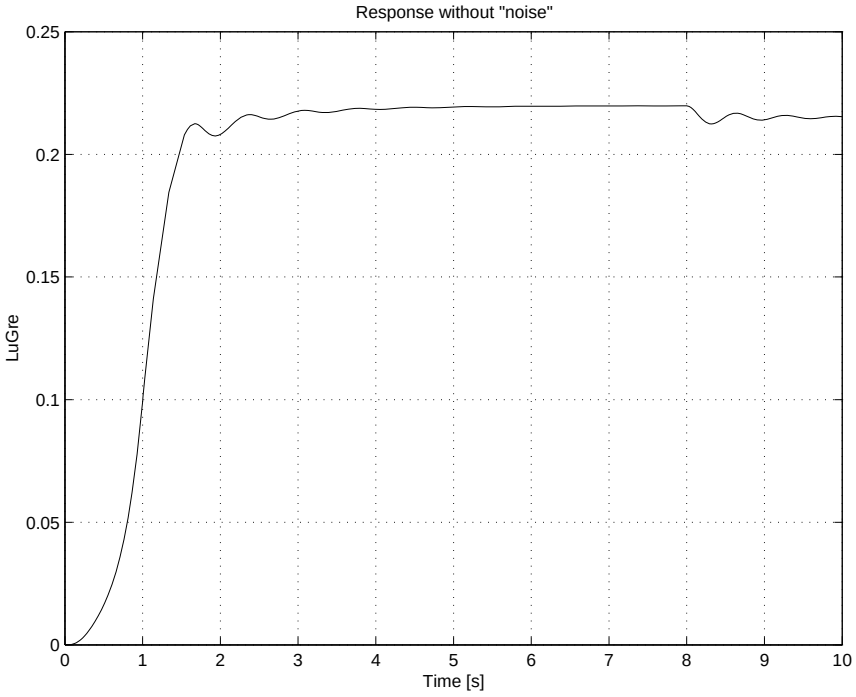


Figure 2.10: Simulation without ”noise”

Chapter 3

Modelling

3.1 DC Servo Motor

An experimental setup of a DC motor will be presented here. The setup is provided by ABB, Billingstad, Norway.

Given the following equations of a DC-motor:

$$L_a \frac{di_a}{dt} = -R_a i_a - K_E \omega_m + u_a \tag{3.1}$$

$$J_m \frac{d\omega_m}{dt} = K_T i_a - T_L - T_f \tag{3.2}$$

$$\frac{d\theta_m}{dt} = \omega_m \tag{3.3}$$

where i_a is the armature current, R_a is the armature resistance, L_a is the armature inductance, u_a is the armature voltage, ω_m is the angular velocity of motor shaft, J_m is the inertia of the motor shaft, T_L is the torque that acts on the motor shaft from the load and θ_m is the motor angle. The constants K_T and K_E are the torque and field constants, where $K_T = K_E$. T_f is the friction torques of the system.

The setup is illustrated in Figure 3.1. An additional load torque is supplied by connecting the motor to a cylindrical transmission shaft by a strap. The interconnection of the motor and the load is assumed to form a rigid system.

An additional equation provides the load torque, which can be inserted into the equations above:

$$J_L \frac{d\omega_L}{dt} = \frac{1}{n} T_L \tag{3.4}$$

Experimental results are presented in chapter 5 concerning static, Lugre and Elasto-Plastic compensation.

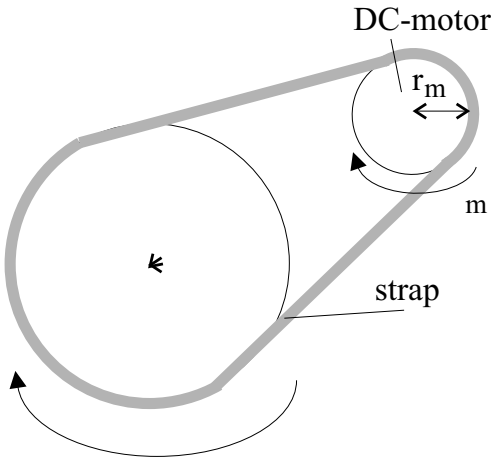


Figure 3.1: Experimental setup of a DC-motor and the load of a friction torque

3.2 Manipulator

The theory in this chapter is based on a study of [16] in addition to the others mentioned below.

3.2.1 Dynamic model of a robot manipulator

The dynamics of a robot manipulator is described by a set of differential equations written in compact form as

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \tag{3.5}$$

where $D(q)\ddot{q}$ is the vector of inertia, $C(q,\dot{q})\dot{q}$ is the vector of Coriolis and centrifugal forces and $g(q)$ is gravitational forces. $\tau \in \mathbb{R}^n$ is the vector of control torques and the vector q describes the motor position.

This plant is described by Euler-Lagrange equations of the form

$$\frac{d}{dt}\left(\frac{\partial L(q,\dot{q})}{\partial \dot{q}}\right) - \frac{\partial L(q,\dot{q})}{\partial q} = \tau \tag{3.6}$$

where $L(q,\dot{q}) := T(q,\dot{q}) - V(q)$ is the Lagrange function, $T(q,\dot{q}) = \frac{1}{2}\dot{q}^T D(q)\dot{q}$, $D(q) = D^T(q) > 0$ is the kinetic energy and $V(q)$ is the potential energy.

The rigid robot manipulator described by the equations of (3.5) has a *passive* mapping from the input torque τ to the angular velocity $\dot{q}$. To verify this, we want to show that there exist a β such that

$$\int_0^t \tau^T \dot{q} ds \geq -\beta, \quad \forall t > 0 \tag{3.7}$$

Consider the total mechanical energy described by the Hamiltonian

$$H(q,\dot{q}) = \frac{1}{2}\dot{q}^T D(q)\dot{q} + \mathbb{U}(q). \tag{3.8}$$

$\mathbb{U}(q)$ is the potential energy due to gravity and $\partial\mathbb{U}/\partial q = g(q)$.

$$\frac{dH}{dt} = \dot{q}^T D(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{D}(q)\dot{q} + \frac{\partial V^T}{\partial q}\dot{q} \tag{3.9}$$

$$= (D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q))^T\dot{q} \tag{3.10}$$

$$= \tau^T\dot{q} \tag{3.11}$$

where we used the property that $\dot{D}(q) - 2C$ is a skew-symmetric matrix, [Craig,1988]. This implies

$$\int_0^t \tau^T\dot{q}ds = H(q(t),\dot{q}(t)) - H(q(0),\dot{q}(0)) \geq -H(q(0),\dot{q}(0)) \tag{3.12}$$

which fulfills the dissipativity condition of equation (3.7).

3.2.2 Two Link Manipulator

In this thesis we are using the two link robot arm of Harry Berghuis [4]. The model of this robot is defined (with zero payload) by

$$D(q) = \begin{bmatrix} 1.02\cos(q_2) + 8.77 & 0.76 + 0.51\cos(q_2) \\ 0.76 + 0.51\cos(q_2) & 0.62 \end{bmatrix} \tag{3.13}$$

$$C(q,\dot{q}) = \begin{bmatrix} -0.51\sin(q_2)\dot{q}_2 & -0.51\sin(q_2)(\dot{q}_1 + \dot{q}_2) \\ 0.51\sin(q_2)\dot{q}_1 & 0 \end{bmatrix} \tag{3.14}$$

$$g(q) = \begin{bmatrix} 74.48\sin(q_1) + 6.174\sin(q_1 + q_2) \\ 6.174\sin(q_1 + q_2) \end{bmatrix} \tag{3.15}$$

In order to simplify the analysis and further work we will assume that the gravity is being compensated by the controller such that we omit $g(q)$ from now.

The model of (3.5) has to be transformed into state space in order to be implemented in a Matlab s-function. We introduce an intermediate variable $x = [q_1 \quad q_2]^T$ to keep track of our variables. In state space we get

$$\dot{x_1} = x_2 \tag{3.16}$$

$$\dot{x_2} = D^{-1}(x_1)\tau - D^{-1}(x_1)C(x_1,x_2)x_2 \tag{3.17}$$

where

$$x_1 = \begin{bmatrix} q_{11} \\ q_{21} \end{bmatrix} \quad \wedge \quad x_2 = \begin{bmatrix} q_{12} \\ q_{22} \end{bmatrix} \tag{3.18}$$

We do the calculations by using the element equivalencies defined by

$$D^{-1}(x_1) = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \tag{3.19}$$

$$C(x_1, x_2) = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \tag{3.20}$$

where each element correspond to the elements in the equations (3.13)-(3.14).

By inserting the above calculations into (3.17) we eventually get our state space model

$$\dot{q_{11}} = q_{12} \tag{3.21}$$
$$\dot{q_{12}} = d_{11}\tau_1 + d_{12}\tau_2 - [(d_{11}c_{11} + d_{12}c_{21})q_{12} + (d_{11}c_{12} + d_{12}c_{22})q_{22}] \tag{3.22}$$
$$\dot{q_{21}} = q_{22} \tag{3.23}$$
$$\dot{q_{22}} = d_{21}\tau_1 + d_{22}\tau_2 - [(d_{21}c_{11} + d_{22}c_{21})q_{12} + (d_{21}c_{12} + d_{22}c_{22})q_{22}] \tag{3.24}$$

which are the equations to be used in the Matlab s-function. Our modification of the two link model results in a two link planar (horizontal plane) model visualized in figure 3.2.

Circle Trajectory

Friction phenomena occurring at low velocities and at sign change can be obtained by tracking a circle. In addition we get a nice visualisation of the friction phenomena when adding friction to the manipulator joints. Before deriving its parametric representation, it is necessary to introduce its significant parameters. Suppose that the circle is specified by assigning:

- the radius r of the circle, defined by $|\overrightarrow{r}|$,
- the center of the circle $\overrightarrow{o} = [o_x, o_y]$,
- the position vector $\overrightarrow{p}$ of a point on the circle.

The end effector of the manipulator is specified when $\overrightarrow{p}$ is known. Link one and two are represented by $\overrightarrow{a_1}$ and $\overrightarrow{a_2}$ respectively. The circle is then specified by

$$\overrightarrow{p} = \overrightarrow{o} + \overrightarrow{r} \tag{3.25}$$

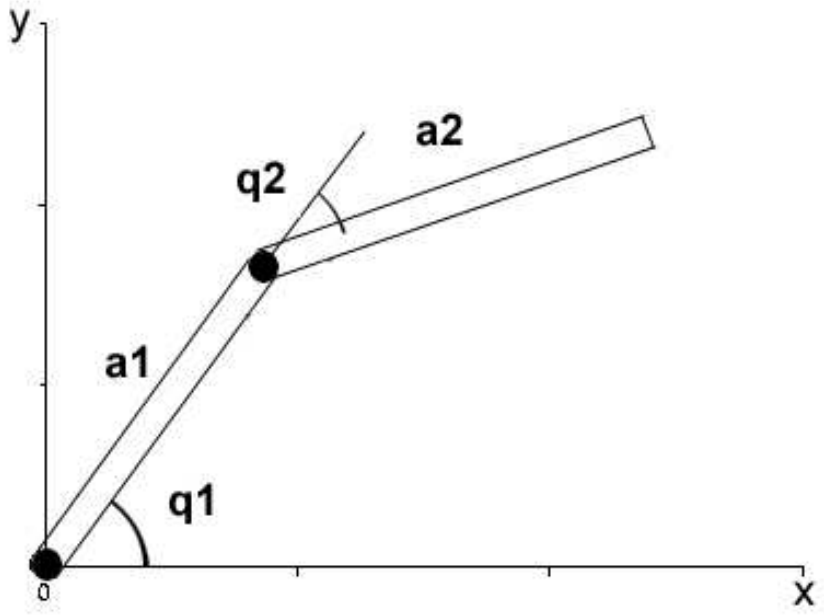


Figure 3.2: Two link planar robot manipulator

which again can be parameterized by

$$p_x = o_x + rcos(\phi) \tag{3.26}$$
$$p_y = o_y + rsin(\phi) \tag{3.27}$$

We now have our reference model in the operational space. Next we will find the joint angles by solving the inverse kinematics.

Inverse Kinematics

Our goal will be to find the joint angles q_1 and q_2 of our manipulator in order to track the circle given by $\overrightarrow{p}$. We also know that $\overrightarrow{p} = \overrightarrow{a_1} + \overrightarrow{a_2}$, which means that the length of the two links represents the position of the end effector. We need to be aware of the constraints

- $\sqrt{p_x^2 + p_y^2} < a_1 + a_2$ gives two solutions,
- $\sqrt{p_x^2 + p_y^2} = a_1 + a_2$ gives one solution,
- $\sqrt{p_x^2 + p_y^2} > a_1 + a_2$ gives no solutions.

where $a_1 = |\overrightarrow{a_1}|$ and $a_2 = |\overrightarrow{a_2}|$.

An algebraic solution technique is used to solve the inverse kinematics. Desirable properties of a solution to inverse kinematics problem are that it should be both *local* and *cyclic*. A

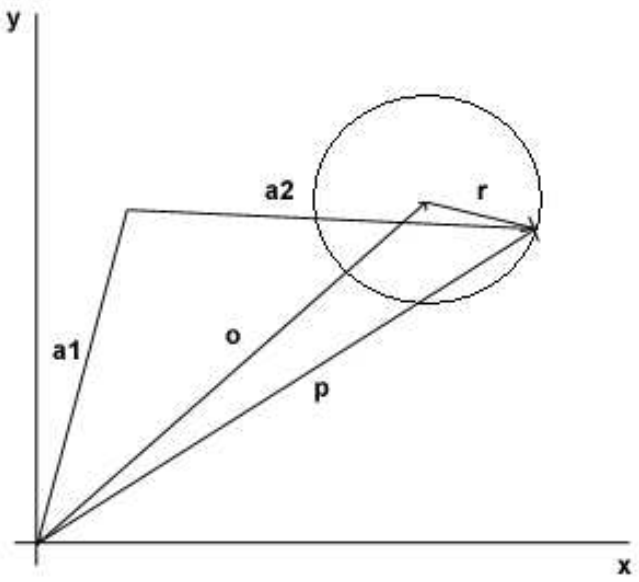


Figure 3.3: Two link manipulator tracking a circle. The joint angles q_1 and q_2 are found by using inverse kinematics

local algorithm is one for which the solution can be determined from local path information by having specified the orientation of the end-effector. From our calculation of $\overrightarrow{p}$ above, we can make an equal solution by representing $\overrightarrow{p}$ by the two robot links $\overrightarrow{a_1}$ and $\overrightarrow{a_2}$ as $\overrightarrow{p} = \overrightarrow{a_1} + \overrightarrow{a_2}$ which we can parameterize as

$$p_x = a_1 \cos(q_1) + a_2 \cos(q_1 + q_2) \tag{3.28}$$
$$p_y = a_1 \sin(q_1) + a_2 \sin(q_1 + q_2) \tag{3.29}$$

where $a_1 = a_2 = 1\text{m}$ according to Berghuis' model. This simplifies our model

$$p_x = \cos(q_1) + \cos(q_1 + q_2) \tag{3.30}$$
$$p_y = \sin(q_1) + \sin(q_1 + q_2) \tag{3.31}$$

By squaring both sides of the equations and summing them up we get

$$p_x^2 + p_y^2 = 2 + 2\cos(q_2) \tag{3.32}$$

where we have used the identity $\cos(A - B) = \cos(A)\cos(B) + \sin(A)\sin(B)$ as resulted in the factor $\cos(q_2)$. Solving the equation with respect to $\cos(q_2)$ and $\sin(q_2)$ we get

$$\cos(q_2) = \frac{p_x^2 + p_y^2 - 2}{2} \tag{3.33}$$
$$\sin(q_2) = \pm\sqrt{1 - \cos^2(q_2)} \tag{3.34}$$

By choosing the positive sign results in the elbow-down posture and the negative sign results in the elbow-up posture. You can chose either one for how its appropriate for your task. The manipulator will be able to track its path with either solution, as long as there exists a solution and in addition there are no obstacles or constraints. We chose the elbow-up posture, which imply negative sign. Hence, the second joint angle can be computed as

$$\vartheta_2 = Atan2(\sin(q_2), \cos(q_2)). \tag{3.35}$$

Having determined ϑ_2, the joint angle ϑ_1 can be obtained by substituting ϑ_2 into (3.31) yielding an algebraic system of two equations in the two unknowns $\sin(q_1)$ and $\cos(q_1)$, where the solution in our case is given by

$$\sin(q_1) = \frac{(1+\cos(q_2))p_y - \sin(q_2)p_x}{p_x^2 + p_y^2} \tag{3.36}$$

$$\cos(q_1) = \frac{(1+\cos(q_2))p_x + \sin(q_2)p_y}{p_x^2 + p_y^2} \tag{3.37}$$

which gives the solution of the first joint angle ϑ_1

$$\vartheta_1 = Atan2(\sin(q_1), \cos(q_1)). \tag{3.38}$$

3.3 Controller Design

When plant dynamics are exactly known, a feedback linearization is an appropriate approach in controlling a manipulator. Feedback linearization amounts to canceling the nonlinearities in a nonlinear system so that the closed-loop dynamics is in a linear form. We will use this technique when compensating for exactly known friction parameters in our system, i.e. a feedback linearization combined with a feed forward friction compensation. Since the LuGre and Elasto-Plastic friction models have a nonmeasurable internal state, z, we also need an observer to estimate it.

Backstepping controller design is a recursive approach divided into several steps, which include Lyapunov design for each step. The resulting lyapunov function is used to prove stability. For each step a virtual control is designed, and the resulting error is transferred into the next step. Backstepping gives the designer the possibility to exploit "good" nonlinearities while "bad" nonlinearities can be dominated e.g. by adding nonlinear damping. One advantage is reduced control power consumption. Additional robustness is obtained. In control systems including friction, there will be some unprecise dynamics present despite the use of advanced friction models.

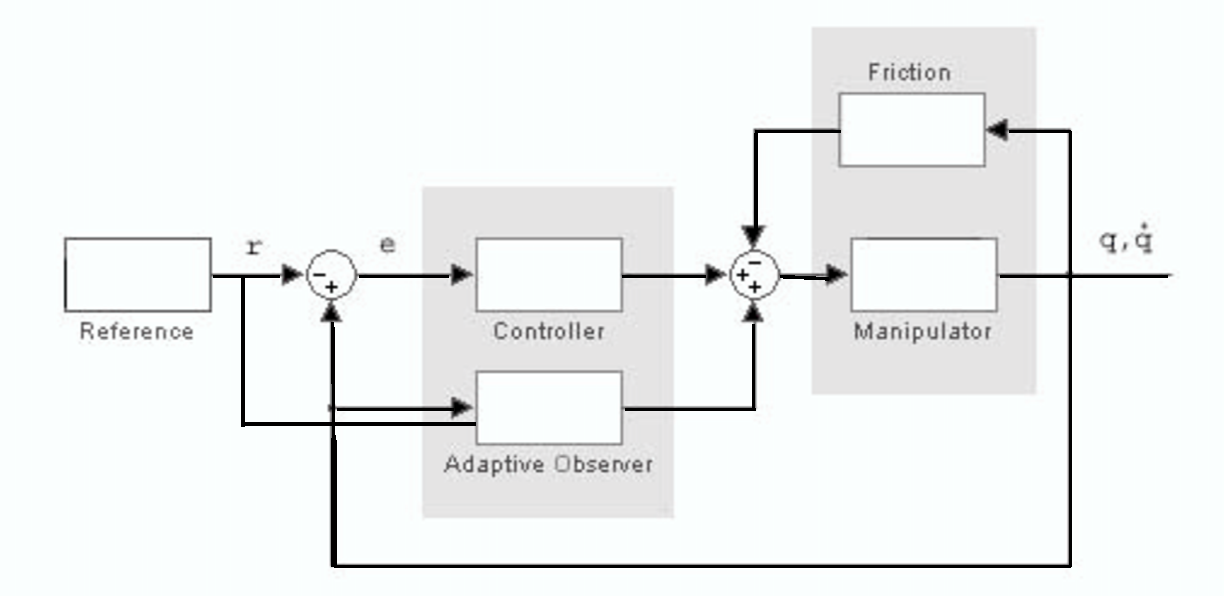


Figure 3.4: Controller system

3.3.1 Observer Design

Given a dynamical system, the observer aims at obtaining an estimate of the current state by only using available measurements. For linear systems, the property of observability,

characterized by the Kalman rank condition, guarantees the possibility to indeed design an observer. In the case of non-linear systems, observability is not enough, basically because this property in general depends on the input of the system. In other words, observability of a nonlinear system does not exclude the existence of inputs for which two distinct initial states cannot be distinguished by using the knowledge of the measured output. This result in the fact that in general, observer gains can be expected to depend on the applied input. Moreover, the existing observers generally tightly depend on some specific structure of the considered system, [15].

From basic control theory we know that if a system is controllable we can place the closed-loop poles wherever we want to. If one has more inputs than "necessary", i.e. all inputs are not needed to make the system controllable, one also has some freedom in placing the eigenvectors. More inputs in the control problem relates to the dual observer problem as more measurements. The more outputs we have, the more freedom we get when designing the observer. The role of the observer is to make full use of the system information in the system outputs, the dual observer takes advantage of the fact that the system can be excited from more than one input, [21].

3.3.2 Adaptive Controller and Parameter Estimation

The presence of parametric uncertainties in dynamic friction models motivates an adaptive approach. While the control system is running, an on-line parameter estimation technique adapts the unknown parameters. There are several parameters in the LuGre model that needs to be known in order to represent the friction force. However, the need of the parameters to converge is not always the issue, actually in a tracking problem the position/velocity-error is desired to converge to zero. It is sufficient that the parameters are bounded. For parameter convergence to occur, a *Persistently Exciting* reference signal is needed.

Definition 3.3.1 *Persistence of Excitation A reference signal* $r(t) = 1 - cos(2\pi\omega t)$ *is Persistently Exciting (PE) if it satisfies*

$$\alpha_1 I \geq \frac{1}{T_0}\int_t^{t+T_0} r(\tau)^2 d\tau \geq \alpha_0 I, \quad \forall t \geq 0 \tag{3.39}$$

for some T_0, α_0, $\alpha_1 > 0$.

We say that r is *sufficiently rich* for identifying the plant if it contains a sufficient number of frequencies to excite all the modes of the plant. Generally speaking for linear systems, the convergent estimation of m parameters require at least $m/2$ sinusoids in the reference signal $r(t)$. However, for the nonlinear case, such simple relation may not be valid.

Chapter 4

Case studies

The performance of tracking control using different friction models will be studied in this chapter. Both *static* and *dynamic* models are presented. It is assumed that the friction parameters are known in section 4.1 and 4.2, while all parameters are unknown in section 4.3. When using known friction parameters we aim to show the difference in performance. In these studies, an Elasto-Plastic friction model is integrated in the manipulator dynamics, so as to simulate the real friction.

For the case of unknown friction parameters the LuGre friction model is studied. Here we also use the LuGre model as the integrated part of the manipulator system. The only assumption made is known stribeck-curve.

Chapter 4.4 present a different approach using neural networks to find the unknown stribeck curve in the LuGre friction model. This approach is a summary of an article presented by [9]. The idea is to find out if it is possible to extend this approach to the elasto-plastic friction model. This will be discussed in chapter 6.

4.1 Control System Using Static Compensation Scheme

Consider the static friction model given as follows:

$$F_f = g(\dot{q})sign(\dot{q}) + F_v\dot{q} \tag{4.1}$$

where $\dot{q}$ is the manipulator velocity vector, $g(\dot{q})$ is the stribeck-curve and F_v is the viscous friction constant. We assume all parameters being known. Recall the manipulator dynamic model in the form:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau - F \tag{4.2}$$

where F represents the real friction, as in this simulation study is represented by the Elasto-Plastic friction model. The desired task is to follow the circle trajectory given in

chapter 3.2.2. A PD position tracking controller is chosen as

$$\tau = D(q)\ddot{q}_d - K_p\tilde{q} - K_d\dot{\tilde{q}} \tag{4.3}$$

where $\tilde{q} = q_d - q$ represents the position error, based up on the assumption that position and velocity are available for measurement. Further the friction torque F will be compensated by the static friction torque F_f. This is feedback linearization and feed forward friction compensation, hence linear control theory provides stability. Results are shown in chapter 5.

4.2 Control System Using Dynamic Compensation Scheme

Difficulties presented by the dynamic friction models includes several unknown parameters and an additional unknown and unmeasurable state, z. In this section we assume that the unknown parameters have been measured off-line. Further we need an observer to provide information about the bristle state. We will show the performance of both the LuGre and the Elasto-Plastic friction models in chapter 5.

4.2.1 The Lund-Grenoble Friction Model

Consider now the problem of tracking an operational space trajectory using the dynamic LuGre friction model. The manipulator dynamic model in the form:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau - F \tag{4.4}$$

where F represents the real friction.

$$\tau = D(q)\ddot{q}_d + \sigma_0\hat{z} + \sigma_1\dot{\hat{z}} + \sigma_2\dot{q} - K_p\hat{q} - K_d\dot{\tilde{q}} \tag{4.5}$$

where the terms including σ_i $(i \in 0, 1, 2)$ are added to represent the LuGre friction. The stability of this controller is the same as for the static case above, knowing from chapter 2.3.4 that the LuGre model is passive. In this control scheme, we can't measure the unknown bristle state z and an observer is needed:

$$\dot{\hat{z}} = \dot{q} - \sigma_0\frac{|\dot{q}|}{g(\dot{q})}\hat{z} \tag{4.6}$$

where $\dot{q}$ is measurable.

4.2.2 The Elasto-Plastic Friction Model

The Elasto-Plastic friction model contain an additional function described by $\alpha(z, \dot{q})$:

$$\alpha(z,\dot{q}) = \begin{cases} 0, & |z| \leq z_{ba}, \\ 0 < \alpha < 1, & z_{ba} < |q| < z_{max}(\dot{q}), \\ 1, & |z| \geq z_{max}(\dot{q}) \end{cases} \tag{4.7}$$

By including this into the bristle state observer, we get the following state observer:

$$\dot{\hat{z}} = \dot{q} - \alpha(\hat{z},\dot{q})\frac{\sigma_0|\dot{q}|}{g(\dot{q})}\hat{z} \tag{4.8}$$

where α is represented by equation (4.8). The total controller is now:

$$\tau = D(q)\ddot{q}_d + \sigma_0\hat{z} + \sigma_1\dot{\hat{z}} + \sigma_2\dot{q} - K_p\hat{q} - K_d\dot{\tilde{q}} \tag{4.9}$$

as in the LuGre case, except for the difference in the $\dot{\hat{z}}$ dynamics.

4.3 Adaptive Friction Compensation

In general, all the friction parameters and internal states are unknown. These parameters can be estimated *on-line* by using an adaptive approach while running a nonlinear observer to observe the unknown bristle state z. The design tool to be used is based on [13] regarding backstepping, while a dual-observer design is motivated by [20]. The dual-observer captures different nonlinear effects of the unmeasurable friction state.

The manipulator equations can now be rewritten in a backstepping point of view:

$$\dot{\mathbf{q}} = \mathbf{v} \tag{4.10}$$

$$\mathbf{D}(\mathbf{q})\dot{\mathbf{v}} = \tau - \mathbf{F} - \mathbf{C}(\mathbf{q},\mathbf{v})\mathbf{v} - \mathbf{g}(\mathbf{q}) \tag{4.11}$$

where $\mathbf{D}(\mathbf{q}) = \mathbf{D^T}(\mathbf{q}) > 0$ is the inertia matrix, $\mathbf{C}(\mathbf{q},\mathbf{v})$ is a matrix of Coriolis and centripetal terms defined in terms of the Christoffel symbols and $\mathbf{g}(\mathbf{q})$ is a vector of gravitational forces and moments. $\mathbf{q} \in \mathbf{R}^n$ is a vector of joint angles, $\mathbf{v} \in \mathbf{R}^n$ is a vector of joint angular rates, $\tau \in \mathbf{R}^n$ is a vector of control torques and $\mathbf{F} \in \mathbf{R}^n$ is a vector of friction torques. The friction torques are represented by the LuGre friction model

$$\dot{\mathbf{z}} = \mathbf{v} - \frac{||\mathbf{v}||}{\mathbf{g_i}(\mathbf{v_i})}\mathbf{z} \tag{4.12}$$

$$\mathbf{F} = \sigma_\mathbf{0}\mathbf{z} + \sigma_\mathbf{1}\dot{\mathbf{z}} + \sigma_\mathbf{2}\mathbf{v} \tag{4.13}$$

where σ_i $i \in \{0, 1, 2\}$ are diagonal matrices. $g_i(v_i)$ is the scalar stribeck curve for each joint $i \in R^n$. By rewriting the friction torques to include the derivatives of the bristle states, we get

$$\mathbf{F} = \sigma_0 \mathbf{z} + \sigma_1 (\mathbf{v} - \frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}} \mathbf{z}) + \sigma_2 \mathbf{v} \tag{4.14}$$

$$= \sigma_0 \mathbf{z} + (\sigma_1 + \sigma_2)\mathbf{v} - \sigma_1 \frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}} \mathbf{z} \tag{4.15}$$

$$= \beta_0 \mathbf{z} + \beta_1 \mathbf{v} - \beta_2 \frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}} \mathbf{z} \tag{4.16}$$

where $\beta_0 = \sigma_0$, $\beta_1 = \sigma_1 + \sigma_2$ and $\beta_2 = \sigma_1$. All parameters and the bristle state are unknown.

Defining the backstepping error in positions $\mathbf{e_1} = \mathbf{q} - \mathbf{q_d}$ where $\mathbf{q_d}$ is a vector of desired positions and $\dot{\mathbf{e}}_1 = \mathbf{v} - \dot{\mathbf{q}}_\mathbf{d}$ is the error in velocities. The virtual control is defined by $\mathbf{v} = \alpha + \mathbf{e_2}$ resulting in the backstepping states

$$\dot{\mathbf{e}}_1 = \alpha + \mathbf{e_2} - \dot{\mathbf{q}}_\mathbf{d} \tag{4.17}$$

$$\mathbf{D}\dot{\mathbf{e}}_\mathbf{2} = D\dot{v} - D\dot{\alpha} \tag{4.18}$$

$$\mathbf{D(q)}\dot{\mathbf{e}}_\mathbf{2} = \tau - \mathbf{F} - \mathbf{C(q, v)v} - \mathbf{g(q)} - \mathbf{D(q)}\dot{\alpha} \tag{4.19}$$

A stabilizing function is chosen as $\alpha = \dot{\mathbf{q}}_\mathbf{d} - \mathbf{k_1 e_1} - \kappa_1 \mathbf{n_1(e_1)e_1}$ where $\mathbf{k_1} > 0$ and $\kappa_1 > 0$ are diagonal matrices at the disposal of the designer. A *nonlinear damping* $\mathbf{n_1(e_1)}$ can in general be introduced to guarantee global boundedness in the absence of adaption, as well as to enhance performance. Now the resulting error states appears as following

$$\dot{\mathbf{e}}_1 = -\mathbf{k_1 e_1} - \kappa_1 \mathbf{n_1(e_1)e_1} + \mathbf{e_2} \tag{4.20}$$

We now chose a Lyapunov function

$$\mathbf{V_1} = \mathbf{e_1^T} \frac{1}{2} \mathbf{e_1} \tag{4.21}$$

and the time derivative

$$\dot{\mathbf{V}}_\mathbf{1} = \mathbf{e_1^T} \dot{\mathbf{e}}_\mathbf{1} \tag{4.22}$$

$$= -\mathbf{e_1^T k_1 e_1} + \mathbf{e_1^T e_2} - \kappa_1 \mathbf{n_1(e_1)e_1^2} \tag{4.23}$$

which will be used in the next step.

In order to design the update laws for the friction parameter estimates, we will depend on the Lyapunov function chosen by

$$\mathbf{V_2} = \mathbf{V_1} + \frac{1}{2} \left(\mathbf{e_2^T} D(q) \mathbf{e_2} + \tilde{\mathbf{z}}_\mathbf{0}^\mathbf{T} \beta_0 \tilde{\mathbf{z}}_\mathbf{0} + \tilde{\mathbf{z}}_\mathbf{1}^\mathbf{T} \beta_2 \tilde{\mathbf{z}}_\mathbf{1} + \tilde{\beta}_\mathbf{0}^\mathbf{T} \mathbf{\Gamma_0^{-1}} \tilde{\beta}_0 + \tilde{\beta}_\mathbf{1}^\mathbf{T} \mathbf{\Gamma_1^{-1}} \tilde{\beta}_1 + \tilde{\beta}_\mathbf{2}^\mathbf{T} \mathbf{\Gamma_2^{-1}} \tilde{\beta}_2 \right) \tag{4.24}$$

and the time derivative

$$\dot{\mathbf{V}}_\mathbf{2} = \dot{\mathbf{V}}_\mathbf{1} + \mathbf{e_2^T} D(q)\dot{\mathbf{e}}_\mathbf{2} + \frac{1}{2}\mathbf{e_2^T}\dot{D}(q)\mathbf{e_2} - \tilde{\beta}_\mathbf{0}^\mathbf{T}\mathbf{\Gamma_0^{-1}}\dot{\hat{\beta}}_\mathbf{0} - \tilde{\beta}_\mathbf{1}^\mathbf{T}\mathbf{\Gamma_1^{-1}}\dot{\hat{\beta}}_\mathbf{1} - \tilde{\beta}_\mathbf{2}^\mathbf{T}\mathbf{\Gamma_2^{-1}}\dot{\hat{\beta}}_\mathbf{2} + \tilde{\mathbf{z}}_\mathbf{0}^\mathbf{T}\beta_\mathbf{0}\dot{\tilde{\mathbf{z}}}_\mathbf{0} + \tilde{\mathbf{z}}_\mathbf{1}^\mathbf{T}\beta_\mathbf{1}\dot{\tilde{\mathbf{z}}}_\mathbf{1} \tag{4.25}$$

Before we can continue, we have to define the dual-observer dynamics as

$$\dot{\hat{\mathbf{z}}}_\mathbf{0} = \mathbf{v} - \frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\hat{z}_0 + \iota_\mathbf{0} \tag{4.26}$$

$$\dot{\hat{\mathbf{z}}}_\mathbf{1} = \mathbf{v} - \frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\hat{z}_1 + \iota_\mathbf{1} \tag{4.27}$$

where ι_0 and ι_1 are additional dynamics to be defined later. Taking $\tilde{\mathbf{z}}_\mathbf{0} = \mathbf{z} - \hat{\mathbf{z}}_\mathbf{0}$ and $\tilde{\mathbf{z}}_\mathbf{1} = \mathbf{z} - \hat{\mathbf{z}}_\mathbf{1}$ as the error dynamics computed as

$$\dot{\tilde{\mathbf{z}}}_\mathbf{0} = -\frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\tilde{z}_0 - \iota_\mathbf{0} \tag{4.28}$$

$$\dot{\tilde{\mathbf{z}}}_\mathbf{1} = -\frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\tilde{z}_1 - \iota_\mathbf{1} \tag{4.29}$$

which can be inserted into the Lyapunov function together with the error dynamics of the system

$$\dot{\mathbf{V}}_\mathbf{2} = -\mathbf{e}_1^T\mathbf{k_1}\mathbf{e}_1 + \mathbf{e}_1^T\mathbf{e}_2 - \mathbf{e}_1^T\kappa_1 n_1(e_1)\mathbf{e}_1 + \frac{1}{2}\mathbf{e}_2^T\dot{D}(q)\mathbf{e}_2 \tag{4.30}$$

$$+ \mathbf{e}_2^T\left(\tau - \mathbf{C(q,v)}\alpha - \mathbf{C(q,v)e_2} - \beta_\mathbf{0}\mathbf{z} - \beta_\mathbf{1}\mathbf{v} + \beta_\mathbf{2}\frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\mathbf{z} - \mathbf{D(q)}\dot{\alpha}\right) \tag{4.31}$$

$$- \tilde{\beta}_\mathbf{0}^\mathbf{T}\mathbf{\Gamma_0^{-1}}\dot{\hat{\beta}}_\mathbf{0} - \tilde{\beta}_\mathbf{1}^\mathbf{T}\mathbf{\Gamma_1^{-1}}\dot{\hat{\beta}}_\mathbf{1} - \tilde{\beta}_\mathbf{2}^\mathbf{T}\mathbf{\Gamma_2^{-1}}\dot{\hat{\beta}}_\mathbf{2} \tag{4.32}$$

$$+ \tilde{\mathbf{z}}_\mathbf{0}^\mathbf{T}\beta_0\left(-\frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\tilde{z}_0 - \iota_\mathbf{0}\right) + \tilde{\mathbf{z}}_\mathbf{1}^\mathbf{T}\beta_1\left(-\frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\tilde{z}_1 - \iota_\mathbf{1}\right) \tag{4.33}$$

Now we can choose the control torques as

$$\tau = \mathbf{D(q)}\dot{\alpha} - \mathbf{e_1} - \mathbf{k_2}\mathbf{e_2} + \mathbf{C(q,v)}\alpha + \mathbf{g(q)} - \kappa_\mathbf{2}\mathbf{n_2(e_2)e_2} + \hat{\beta}_\mathbf{0}\hat{\mathbf{z}}_\mathbf{0} + \hat{\beta}_\mathbf{1}\mathbf{v} - \hat{\beta}_\mathbf{2}\frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\hat{\mathbf{z}}_\mathbf{1} \tag{4.34}$$

Inserting the control torques and organizing the Lyapunov function, we get

$$\dot{\mathbf{V}}_\mathbf{2} = -\mathbf{e}_1^T\mathbf{k_1}\mathbf{e}_1 - \mathbf{e}_2^T\mathbf{k_2}\mathbf{e}_2 - \mathbf{e}_1^T\kappa_1 n_1(e_1)\mathbf{e_1} - \mathbf{e}_2^T\kappa_2 n_2(e_2)\mathbf{e}_2 + \frac{1}{2}\mathbf{e}_2^T\dot{D}(q)\mathbf{e}_2 - \mathbf{e_2^T C(q,v)e_2} \tag{4.35}$$

$$- \tilde{\beta}_\mathbf{0}\left(\mathbf{e_2}\hat{\mathbf{z}}_\mathbf{0} + \mathbf{\Gamma_0^{-1}}\dot{\hat{\beta}}_\mathbf{0}\right) - \tilde{\beta}_\mathbf{1}\left(\mathbf{ve_2} + \mathbf{\Gamma_1^{-1}}\dot{\hat{\beta}}_\mathbf{1}\right) + \tilde{\beta}_\mathbf{2}\left(\frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\hat{\mathbf{z}}_\mathbf{1}\mathbf{e_2} - \mathbf{\Gamma_2^{-1}}\dot{\hat{\beta}}_\mathbf{2}\right) \tag{4.36}$$

$$- \beta_\mathbf{0}\tilde{\mathbf{z}}_\mathbf{0}(\mathbf{e_2} + \iota_\mathbf{0}) + \beta_\mathbf{2}\tilde{\mathbf{z}}_\mathbf{1}\left(\frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\mathbf{e_2} - \iota_\mathbf{1}\right) \tag{4.37}$$

$$- \frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\beta_\mathbf{0}\tilde{\mathbf{z}}_\mathbf{0}^\mathbf{2} - \frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\beta_\mathbf{2}\tilde{\mathbf{z}}_\mathbf{1}^\mathbf{2} \tag{4.38}$$

where $\tilde{\beta}_i = \beta_i - \hat{\beta}_i$ and by using the fact that $\mathbf{e_2^T}(\frac{\mathbf{1}}{\mathbf{2}}\dot{\mathbf{D}}(\mathbf{q}) - \mathbf{C}(\mathbf{q},\mathbf{v}))\mathbf{e_2} = \mathbf{0}$, choosing update laws and observer dynamics as

$$\dot{\hat{\beta}}_0 = -\mathbf{\Gamma_0 e_2 \hat{z}_0} \tag{4.39}$$

$$\dot{\hat{\beta}}_1 = -\mathbf{\Gamma_1 e_2 v} \tag{4.40}$$

$$\dot{\hat{\beta}}_2 = \mathbf{\Gamma_2}\frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\mathbf{e_2\hat{z}_1} \tag{4.41}$$

$$\iota_0 = -\mathbf{e_2} \tag{4.42}$$

$$\iota_1 = \frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\mathbf{e_2} \tag{4.43}$$

we obtain

$$\dot{\mathbf{V}}_\mathbf{2} = -\mathbf{e}_1^T\mathbf{k_1 e}_1 - \mathbf{e}_2^T\mathbf{k_2 e}_2 - \mathbf{e}_1^T\kappa_1 n_1(e_1)\mathbf{e_1} - \mathbf{e}_2^T\kappa_2 n_2(e_2)\mathbf{e}_2 - \frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\beta_\mathbf{0}\tilde{\mathbf{z}}_\mathbf{0}^\mathbf{2} - \frac{||\mathbf{v}||}{\mathbf{g_i(v_i)}}\beta_\mathbf{2}\tilde{\mathbf{z}}_\mathbf{1}^\mathbf{2} \tag{4.44}$$

The expression for $\dot{\alpha}$ which appear in the controller torque τ is computed by taking the time derivative of α along the trajectories of $\dot{\mathbf{q}}_\mathbf{d}$ and $\mathbf{e_1}$, hence

$$\dot{\alpha} = \frac{\partial\alpha}{\partial\dot{q}_d}\ddot{q}_d - \frac{\partial\alpha}{\partial e_1}\dot{e}_1 \tag{4.45}$$

$$= \ddot{q}_d - \frac{\partial\alpha}{\partial e_1}(v - \dot{q}_d) \tag{4.46}$$

The desired state q_d is assumed to be smooth such that $\dot{q}_d$ and $\ddot{q}_d$ exist.

To prove that the system is asymptotically stable and that the position tracking error converges to zero asymptotically, we utilize the Lyapunov function and the closed loop system dynamics. Using Fact 1 in appendix and choosing the constants k_1, k_2, κ_1 and κ_2 all positive definite we obtain $\dot{V}_2 \leq 0$. Combining this with the fact that the Lyapunov function V_2 in (4.3) is positive definite, we conclude that V_2 is bounded, which means that its components e_1, e_2, $\tilde{\beta}_0$, $\tilde{\beta}_1$, $\tilde{\beta}_2$, $\tilde{z}_0$ and $\tilde{z}_1$ are all bounded. Of special interest we have $e_1, e_2 \in L_\infty$.

Since the friction torque parameters β_i are bounded unknown constants and by definition $\tilde{\beta}_i = \beta_i - \hat{\beta}_i$, we can see that the parameters $\tilde{\beta}_i$ are also bounded signals. The position tracking error $\mathbf{e_1} = q - q_d$ is bounded, and we assume q_d to be bounded, then also the motor positions $\mathbf{q}$ are bounded. From the expression of the stabilizing function α and the fact that $\dot{q}_d$ and e_1 are bounded signals, we can see that α is bounded. Combining this with the fact that $\mathbf{v} = \alpha + \mathbf{e_2}$ is bounded, we conclude that the motor angular velocity $\mathbf{v}$ is a bounded signal. Considering the friction dynamics in equation (4.13), we know that the friction dynamics is bounded since the angular velocity $\mathbf{v}$ is bounded. This also mean that the friction state estimates $\hat{z}_0$ and $\hat{z}_1$ are bounded since $\tilde{z}_0 = z - \hat{z}_0$ and $\tilde{z}_1 = z - \hat{z}_1$

are bounded due to the Lyapunov function. The boundedness of the control input τ is apparent from its expression in (4.34) and the fact that all the signals in the expression are globally bounded.

In particular we have $e_1, e_2 \in L_\infty$ and $\dot{e}_1, \dot{e}_2 \in L_\infty$ and it is seen from V_2 that $e_1, e_2 \in L_2$. We conclude that the error signals e_1 and e_2 converge to zero asymptotically.

4.4 Neural Network Approach

In the adaptive case treated above we have assumed the Stribeck curve being known. In this section we will point out some ideas on solving this as proposed in [9] using Neural Networks (NN). NN can be used to parameterise the unknown nonlinear characteristic function of the dynamic friction model. The first case discussed in [9] is by assuming all friction parameters in the LuGre model as known. Further they considered the unknown characteristic function $\alpha(x, \dot{x})$ and proposed using a neural network to parameterise it:

$$\alpha(\dot{x}) = \frac{\sigma_0}{F_c + (F_s - F_c)e^{-(\dot{x}/\dot{x}_s)^2}} \tag{4.47}$$

which yields

$$\dot{z} = \dot{x} - \alpha(\dot{x})|\dot{x}|z \tag{4.48}$$

$$F = \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 \dot{x} \tag{4.49}$$

A NN is taken as a function approximator which emulates a given nonlinear function up to a small error tolerance. It has been proven that any continuous functions can be uniformly approximated by a linear combination of Gaussian radial basis function (RBF) if the size is large enough. The RBF NN is a particular network architecture and can be described as

$$f_{nn}(W, x, \dot{x}) = W^T S(x, \dot{x}) \tag{4.50}$$

where x, $\dot{x}$ are the input variables, $W \in R^l$ is the weight vector, and

$$S(x, \dot{x}) = [s_1(x, \dot{x}), s_2(x, \dot{x}), \cdots, s_l(x, \dot{x})]^T \quad \in \quad R^l \tag{4.51}$$

is the basis function vector having the form of

$$s_i(x, \dot{x}) = exp\left(-\frac{(x - \mu_{1i})^2 + (\dot{x} - \mu_{2i})^2}{\sigma^2}\right), \quad i = 1, \cdots, l \tag{4.52}$$

with $\sigma \in R$ being the variance and $[\mu_{1i}, \mu_{2i}]^T \in R^2$ being the centre vector. Other NN can also be used without any difficulty.

Since this scheme assume the friction parameters σ_0, σ_1 and σ_2 as known we will not reproduce the scheme, but only note that the tracking error converged to zero and that all the signals in the closed-loop were bounded.

In the second case a controller is designed for full set of unknown parameters. This is done by separating the viscous friction parameter into one part, and the terms containing the bristle state z in a second part. The second part is bounded by a function independent of $z(t)$. Then a RBF NN is applied to approximate this unknown bounding function. Based on Lyapunov synthesis, adaption algorithms for both the NN weights and the unknown system and friction parameters are presented. The friction force is now yielding:

$$F = \theta\dot{x} + F_z(x, \dot{x}, z) \tag{4.53}$$

where $\theta = \sigma_1 + \sigma_2$, $\theta\dot{x}$ represents viscous friction force and $F_z(x, \dot{x}, z) = \sigma_0 z - \sigma_1\alpha(x, \dot{x})|\dot{x}|z$ is the dynamic friction force which depends on z.

Proposition 4.4.1 *There exist positive constants α_{min} and α_m such that $0 < \alpha_{min} \leq \alpha(x, \dot{x}) \leq \alpha_m, \forall \quad (x, \dot{x}) \in R^2$*

Lemma 4.4.2 *If $|z(0)| \leq 1/\alpha_{min}$ then$|z(t)| \leq 1/\alpha_{min}, \quad \forall t \geq 0$.*

From the **lemma** we know that F_z is bounded by

$$|F_z(x, \dot{x}, z)| = |(\sigma_0 - \sigma_1\alpha(x, \dot{x}))||z(t)| \tag{4.54}$$

$$\leq \frac{\sigma_0 + \sigma_1\alpha(x, \dot{x})}{\alpha_{\min}} = F_{zm}(x, \dot{x}) \tag{4.55}$$

where $F_{zm}(x, \dot{x})$ is the bounding function of $F_z(x, \dot{x}, z)$ and is *independent* of the unmeasurable internal friction state z. In sequel, an RBF NN is applied to approximate $F_{zm}(x, \dot{x})$, and similarly there exist the following function approximation

$$F_{zm}(x, \dot{x}) = W^{*T}S(x, \dot{x}) + \epsilon \tag{4.56}$$

with W^* being the optimal weight vector, and the NN approximation error ϵ being bounded by a small positive constant ϵ_d, i.e., $|\epsilon| \leq \epsilon_d$.

The servo mechanism studied in [9] is described by

$$m\ddot{x} + F = u \tag{4.57}$$

substituting friction dynamics (4.49) into system equation, we have

$$m\ddot{x} = u - \theta\dot{x} - F_z(x, \dot{x}, z) \tag{4.58}$$

Defining $e = x - x_d$, $\dot{x}_r = \dot{x}_d - \lambda e$ and $r = \dot{e} + \lambda e$, where $\lambda > 0$ and r is the filtered tracking error. Then the tracking error dynamics is transformed into

$$m\dot{r} = u - \theta\dot{x} - F_z(x, \dot{x}, z) - m\ddot{x}_r \tag{4.59}$$

Considering the following controller

$$u = -c_1 r + \hat{\theta}\dot{x} + \hat{m}\ddot{x}_r - \hat{F}_{zm}(x, \dot{x})sgn(r) - ksgn(r) \tag{4.60}$$

where constant $c_1 > 0$, $\hat{\theta}$ and $\hat{m}$ are the estimates of unknown θ and m respectively, $\hat{F}_{zm}(x, \dot{x}) = W^T S(x, \dot{x})$ is the RBF NN approximation of function bound $F_{zm}(x, \dot{x})$ and $k > \epsilon_d$.

Substituting (4.59) into (4.60) yields

$$\begin{aligned} m\ddot{x} =& - c_1 r - \tilde{\theta}\dot{x} - \tilde{m}\ddot{x}_r - \hat{F}_{zm}(x, \dot{x})sgn(r) - ksgn(r) - \theta\dot{x} && (4.61)\\ & - F_z(x, \dot{x}, z) && (4.62) \end{aligned}$$

where $(\tilde{*}) = (*) - (\hat{*})$ denotes the unknown parameter estimation errors. Further manipulation results in the following equation

$$\begin{aligned} m\ddot{x} =& - c_1 r - \tilde{\theta}\dot{x} - \tilde{m}\ddot{x}_r + \tilde{W}^T S(x, \dot{x}) + \epsilon sgn(r) - (k - \epsilon)sgn(r) - F_z(x, \dot{x}, z) && (4.63)\\ & - F_{zm}(x, \dot{x})sgn(r) - \tilde{m}\ddot{x}_r && (4.64) \end{aligned}$$

where $\tilde{W} = W^* - W$. Details in mathematical manipulations can be seen in the article.

Considering the closed-loop system consisting of system (4.57) with dynamic friction given by LuGre dynamics, and adaptive controller (4.60). If the parameters $\hat{\theta}$, $\hat{m}$ and NN weight W are updated by

$$\dot{\hat{\theta}} = -\eta_\theta \dot{x} r \tag{4.65}$$

$$\dot{\hat{m}} = -\eta_m \ddot{x}_r r \tag{4.66}$$

$$\dot{W} = \Gamma S(x, \dot{x})|r| \tag{4.67}$$

where η_θ and η_m are positive constants, $\Gamma = \Gamma^T > 0$ is dimensionally compatible constant matrix, then the tracking error converges to zero and all the signals in the closed-loop are bounded. The proof is seen in [9].

Chapter 5

Results

All simulations were performed by using Matlab/SIMULINK 6.1. It should be noted that there are some differences between 6.0 and 6.1. Some of the source code didn't apply to 6.0.

The objective of these simulations are as follows:

1. Illustrate the performance of *static* vs. *dynamic* compensation schemes on esperimental results.

2. and on simulation studies concerning known and unknown friction parameters.

This chapter is divided into three sections. First we present the experimental results in section 5.1. Next we present simulation results when assuming known friction parameters in section 5.2, and finally we present the adaptive scheme in section 5.3. All results are discussed in chapter 6.

5.1 Experimental Setup

Experimental results from laboratory at ABB, Billingstad, were compared to a DC model with both a static friction model and a dynamic friction model. Friction parameters have been adjusted to fit the experimental result. The reference signal represents a sinus velocity.

It should be noted that these results were not performed by me so that I had no control on the specific setup. The elastic strap connecting the motor to the shaft should probably not have been there for this experiment. Anyway, the results provides a nice visualization of the different friction models.

5.1.1 Experimental environment

The following parameters were used in the simulation model of the DC motor

L_a	$= 1.4 \cdot 10^{-3}$	F_s	$= 1.5$
R_a	$= 0.6$	ω_s	$= 0.0001$
J_m	$= 3.66 \cdot 10^{-5}$		
K_T	$= 0.056$	σ_0	$= 200.0$
K_E	$= K_T$	σ_1	$= 0.04$
F_c	$= 0.1$	σ_2	$= 0.045$

The system was first simulated by using Coloumb and viscous friction forces. Fixed step size was chosen using the Runge-Kutta 4 solver. Another simulation was performed using the dynamic LuGre friction model. Variable step size and ODE4.5 was chosen.

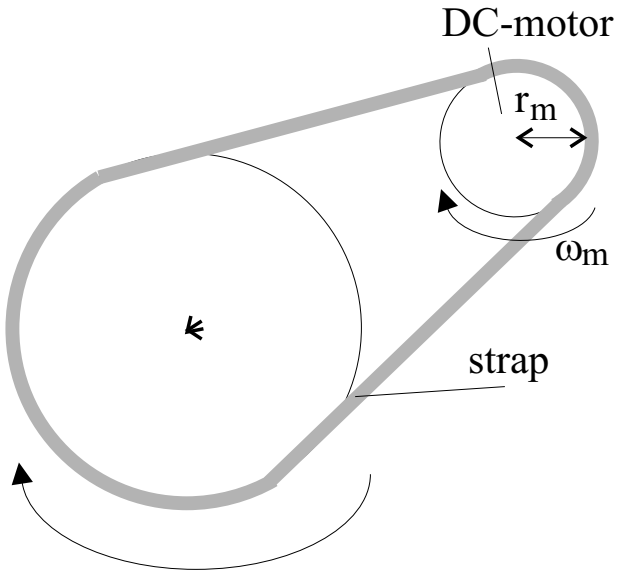


Figure 5.1: Experimental setup of a DC-motor and the load of a friction torque

5.1.2 Experimental Results

Coloumb and viscous Friction

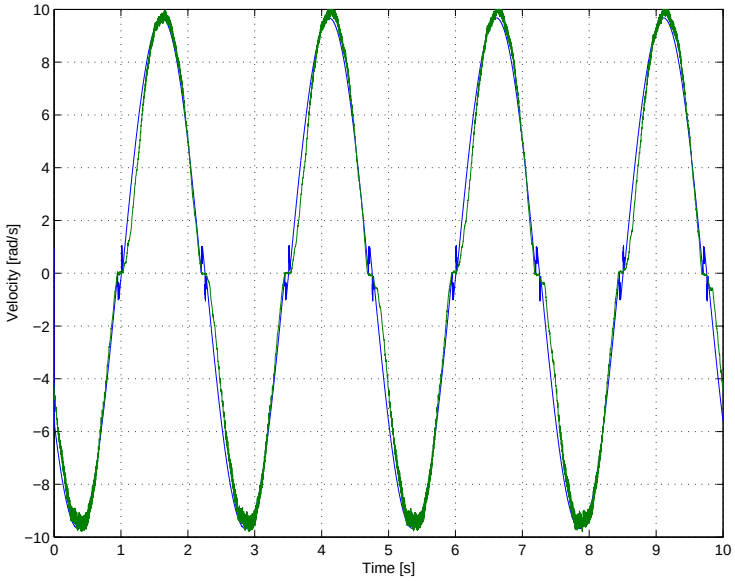


Figure 5.2: Experimental result and a dynamic DC model including static friction

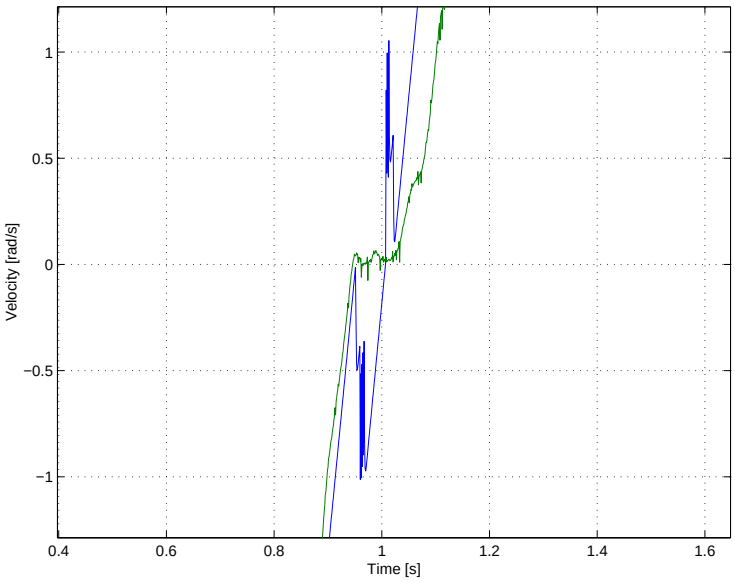


Figure 5.3: A zoom about zero velocity from figure 5.2

LuGre Friction

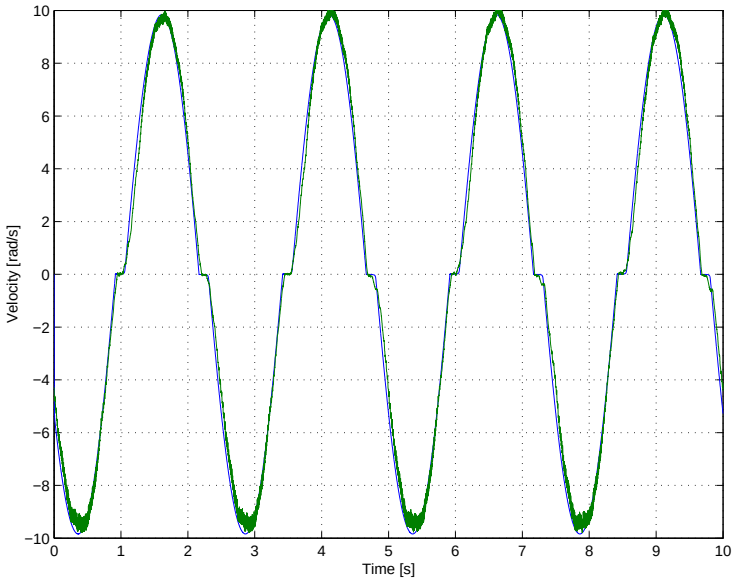


Figure 5.4: Experimental result and a dynamic DC model including LuGre friction

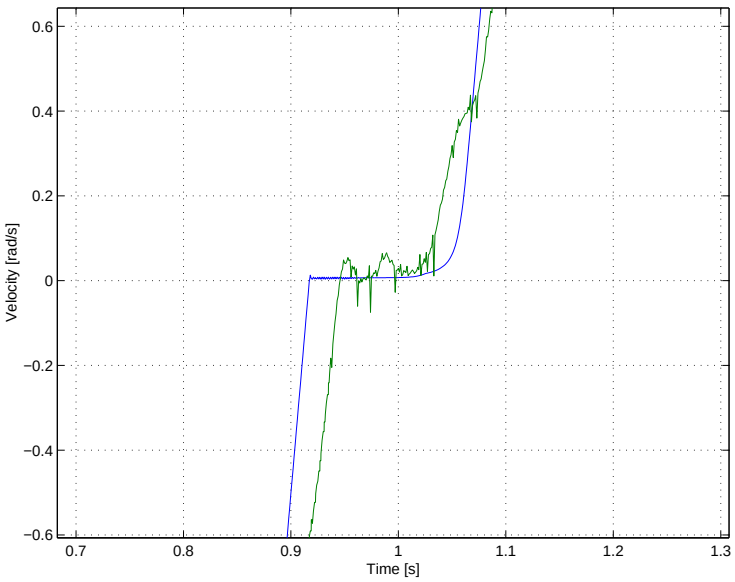


Figure 5.5: A zoom about zero velocity from figure 5.4

Elasto-Plastic Friction

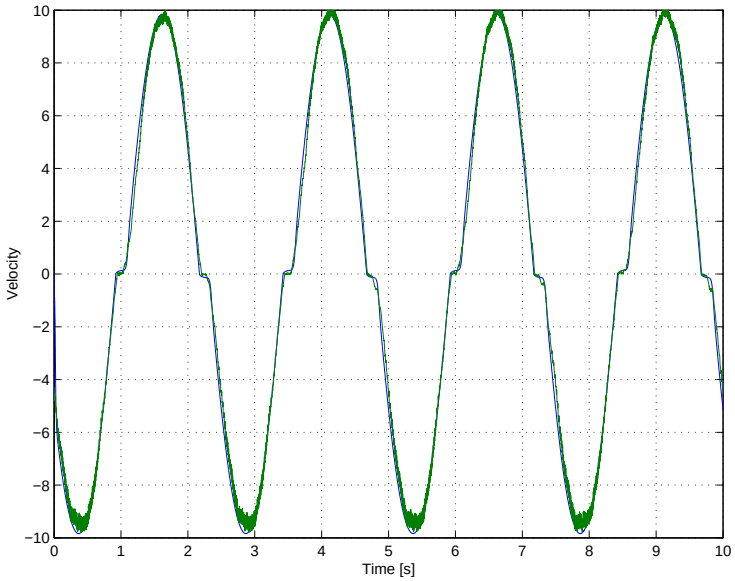


Figure 5.6: Experimental result and a dynamic DC model including Elasto-Plastic friction

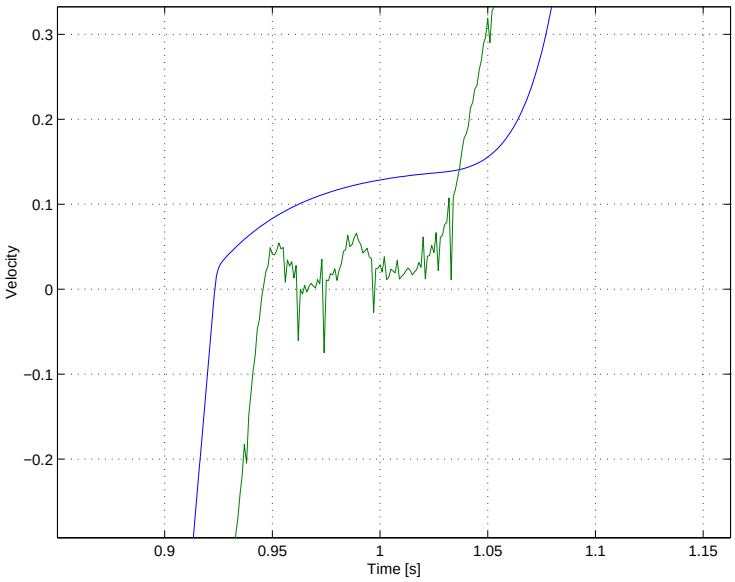


Figure 5.7: A zoom about zero velocity from figure 5.6

5.2 Simulation Results

A PD tracking controller is used with $K_p = 40$ and $K_d = 25$.

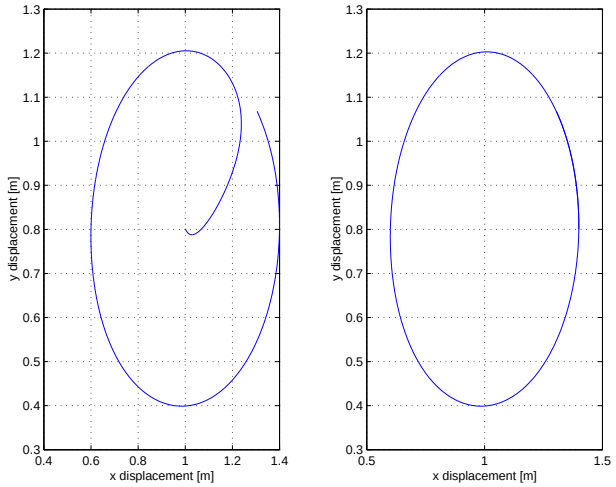


Figure 5.8: Manipulator tracking circle without friction. Left: Manipulator starts outside trajectory with zero velocity at $t_0 = 0$. Right: Manipulator following trajectory with correct velocities and position at $t_0 = 0$

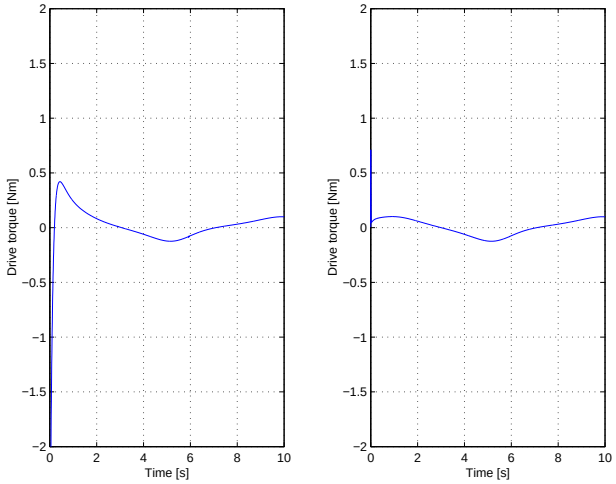


Figure 5.9: Drive torque. Left: Manipulator starts outside trajectory with zero velocity at $t_0 = 0$. Right: Manipulator following trajectory with correct velocities and position at $t_0 = 0$

System including Elasto-Plastic friction

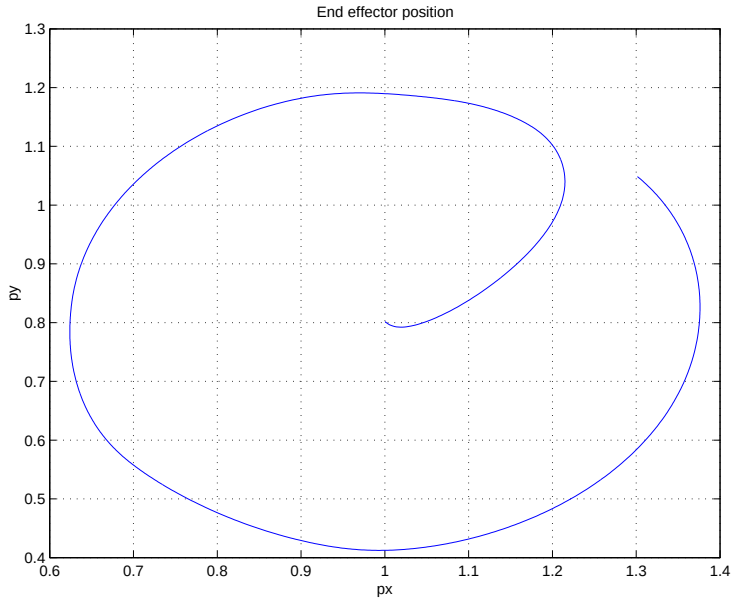


Figure 5.10: Circle trajectory including friction in manipulator joints

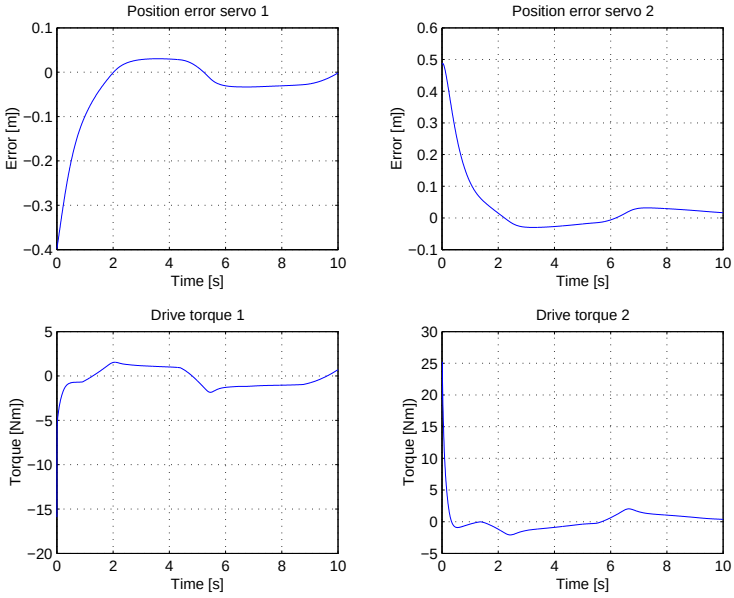


Figure 5.11: Upper left: Position error link 1, Upper right: Position error link 2, Lower left: Drive torque link 1 and Lower right: Drive torque link 2

5.2.1 Static Compensation

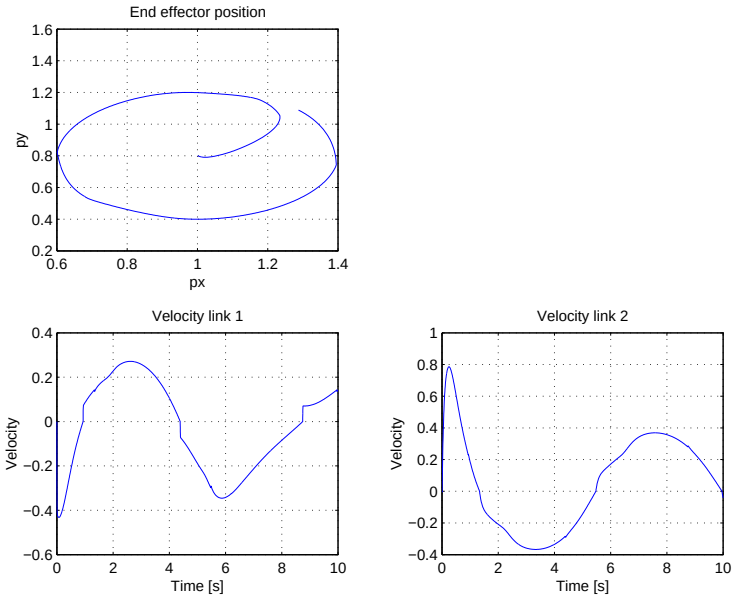


Figure 5.12: Circle trajectory including friction in manipulator joints

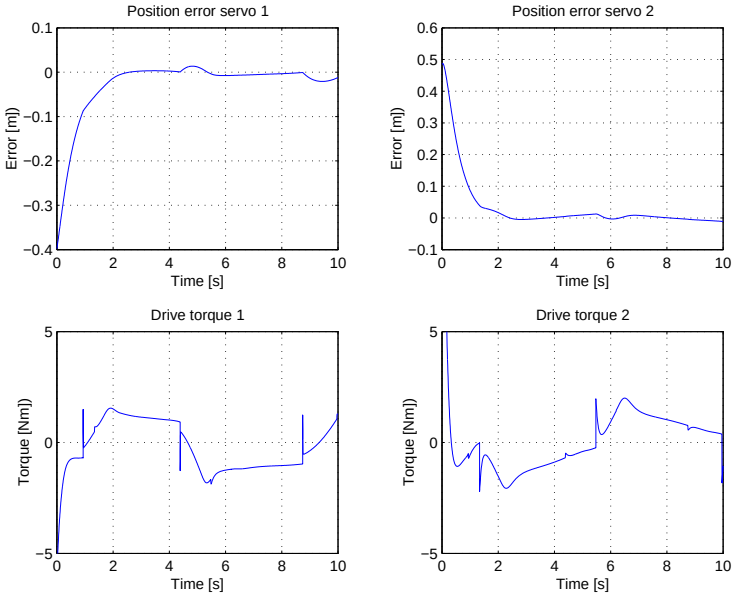


Figure 5.13: Upper left: Position error link 1, Upper right: Position error link 2, Lower left: Drive torque link 1 and Lower right: Drive torque link 2

5.2.2 LuGre Compensation

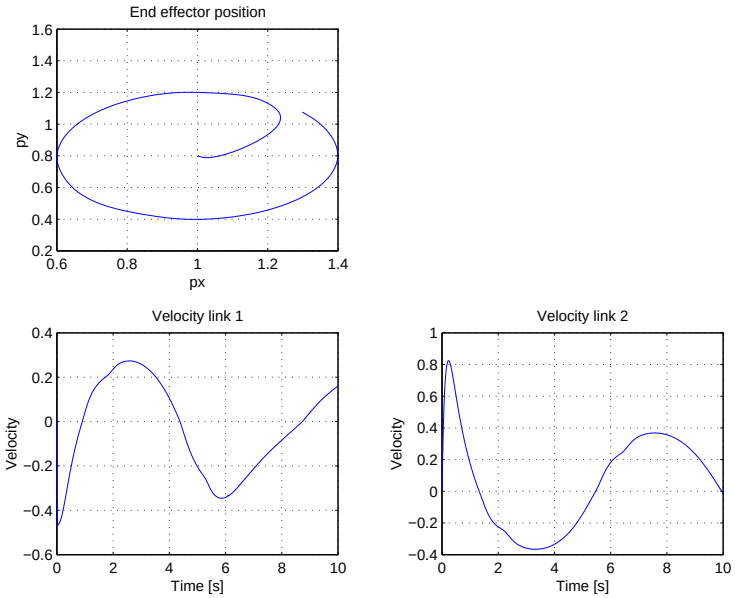


Figure 5.14: Circle trajectory including friction in manipulator joints

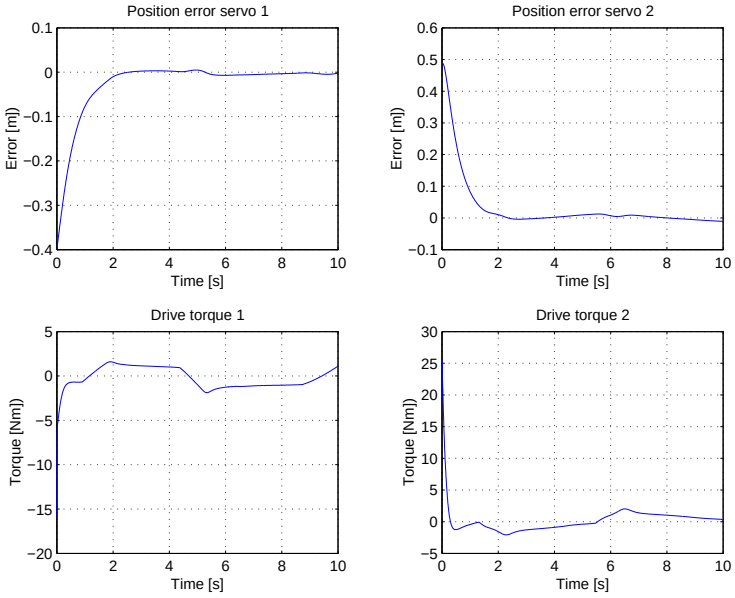


Figure 5.15: Upper left: Position error link 1, Upper right: Position error link 2, Lower left: Drive torque link 1 and Lower right: Drive torque link 2

5.2.3 Elasto-Plastic Compensation

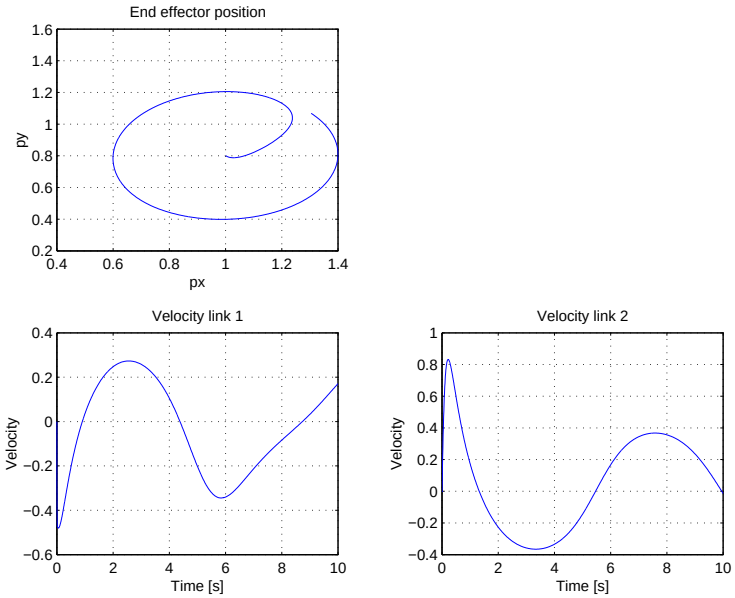


Figure 5.16: Circle trajectory including friction in manipulator joints

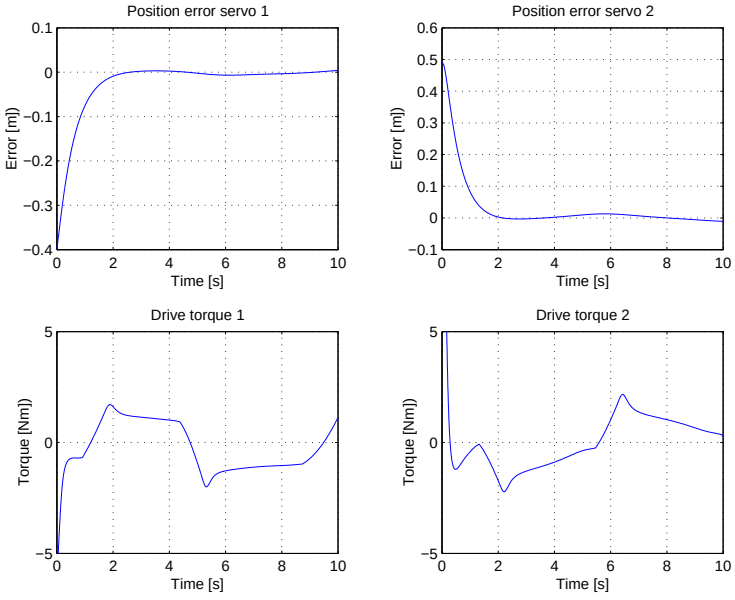


Figure 5.17: Upper left: Position error link 1, Upper right: Position error link 2, Lower left: Drive torque link 1 and Lower right: Drive torque link 2

5.3 Adaptive Friction Compensation

A PE signal $q_d(t) = 1 - cos(0.1t)$ is used to excite the system. This is easy to derivate analytically, and gives no numerical errors.

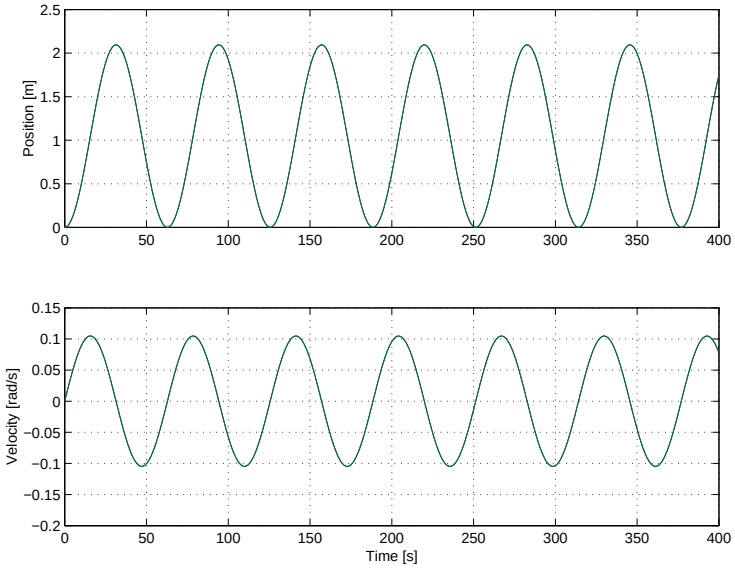


Figure 5.18: Reference signals. Upper: Position, Lower: Velocity

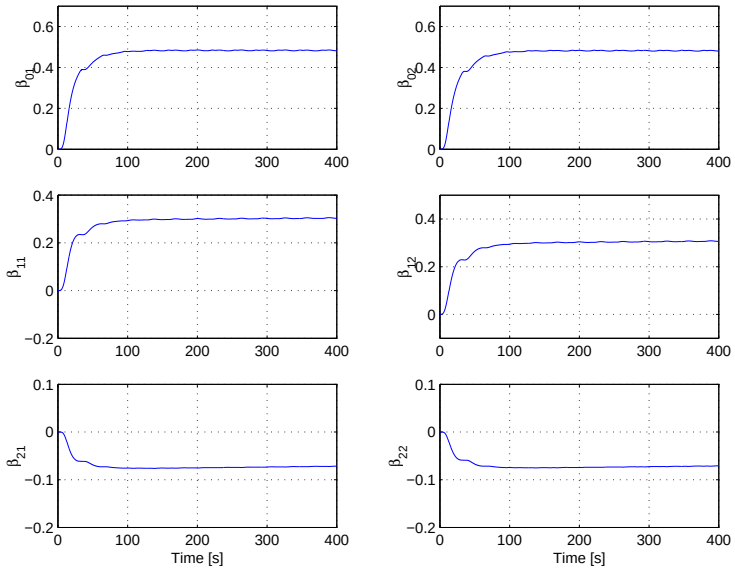


Figure 5.19: Friction parameters. Left column: Joint one, Right column: Joint two

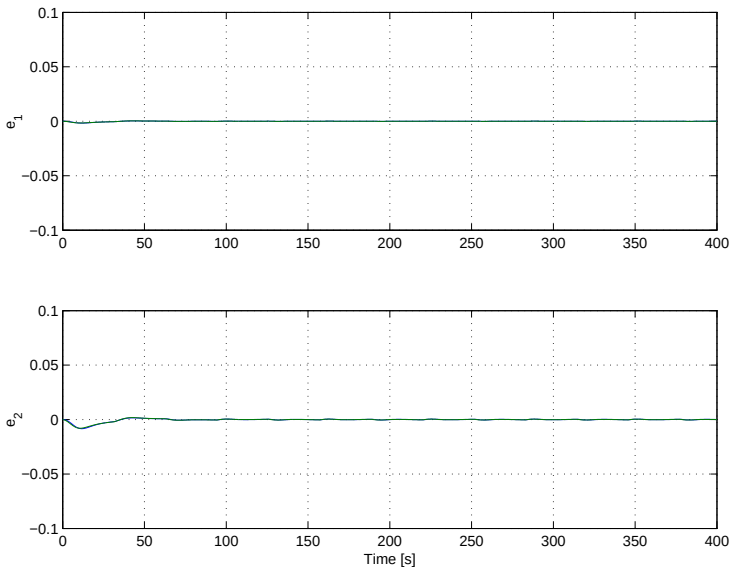


Figure 5.20: Tracking error in e_1 and e_2

Controller gains were selected as follows

$$\lambda = 5 \tag{5.1}$$
$$k_1 = \lambda^2 \tag{5.2}$$
$$k_2 = 2\lambda \tag{5.3}$$

for each joint, wheras the adaptive gains

$$\gamma_0 = 10 \tag{5.4}$$
$$\gamma_1 = 20 \tag{5.5}$$
$$\gamma_2 = 6 \tag{5.6}$$

for each joint.

Chapter 6

Discussions

Friction in mechanical control systems is undesirable and a study of different friction phenomena has provided us with insight which can be used in modelling and control of such systems. The differences between *static* and *dynamic* friction models can be essential in control systems operating at low velocities, and when crossing zero velocity. From the presentation in chapter 2, it is clear that high-precision pointing and tracking applications are dominated by presliding. Dahl was the first to attempt to model presliding for control applications. The LuGre model was designed to extend Dahl's model to include other effects, such as those associated with the sliding of lubricated contacts. Dupont has shown that drift is due to the fact that presliding displacement in the Dahl and LuGre models always include a plastic (irreversible) component. To minimize drift, he presented a class of single-state models in which presliding is elasto-plastic, i.e. under loading the displacement is first purely elastic (reversible) before transitioning to plastic (irreversible).

A closer investigation of the drift shown by Dupont is seen in Figure 2.9. We tried to recreate his simulation study so to be able to search for the main source of the observed drift. The input signal consist of a ramp function from $t_0 = 0 \leq 1$ and a constant signal from about $1 < t_1 \leq 1.8$, a "noise" sinus signal from $1.8 < t_2 \leq 8$ and a zero signal until 10 seconds. The sinus signal represents noise. The two graphs below the input signal is the LuGre and Elasto-Plastic friction responses respectively. The main difference is apparent when $t \in t_2$ where it is seen that the LuGre response is drifting away. In Figure 2.10 we have a simulation without the sinus "noise" in t_2. We observe that the drift is hardly noticed. In practical applications noise will most probably be present, but as seen in the mentioned figures, the drift size or numerical value is quite small compared to the elapsed time. The complexity of implementing the Elasto-Plastic model in a control system makes the LuGre model more attractive considering dynamic models. In addition the Elasto-Plastic model is mainly based on simulation studies and the presented ideas are not yet confirmed experimentally.

6.1 Experimental Results

Figure 5.2 shows the experimental data representing a sinus signal. At each peak the experimental data are noisy. A Coulomb with viscous friction i fitted to the data. We observe that the simulated static model has a discontinuity when crossing zero velocity, while the real system is possessing a stic behaviour. This becomes more obvious in the zoomed Figure 5.3, which is zoomed about zero velocity.

The difficulties caused by Coulomb's friction model in rigid-body dynamic simulation is among others that the friction force isn't smooth during rolling-sliding transitions. As a result of this, some researchers introduced the $\tanh(\dot{x})$ which is making a smooth transition. This is however not reflecting the important friction phenomena when crossing zero velocity. In applications dealing with high velocities and where its not critical to operate precise, the $\tanh(\dot{x})$ solution could apply.

Using the same experimental results, but applying the LuGre friction model we observe improved results as shown in Figure 5.4. A closer look when crossing zero velocity is seen in Figure 5.5 and it demonstrates the advantage of the dynamic friction model. An explanation of the oscillations in the experimental data could be due to elasticity in the strap coupling. An experimental study dedicated to the observation of friction phenomena would have been more appropriate and should be explored further.

Surprisingly we observe a biased signal in the case of the Elasto-Plastic model in Figures 5.6 and 5.7. Notice the scaling of the axis compared to the LuGre case. The bias could be explained due to the elasticity of the strap. Again we are convinced that a dedicated study should be performed. As mentioned above, the Elasto-Plastic friction model has not been confirmed experimentally.

A concluding remark is that the static friction models are performing bad at low velocities.

6.2 Simulation Studies

A robotic manipulator operating precisely has been investigated by simulation studies. The end-effector is required to form a circle, hence position tracking controller in addition to friction compensation were simulated using both static and dynamic friction models.

The graphical presentations of the circle trajectory are somewhat confusing. They are compressed, or stretched so that they visualizes as an ellipse like shape. This is due to presentation error, so that all circle trajectories shall be thought of as circular.

The reference signal is computed numerically using Maple and manually fitting of the derivatives. This ended up giving an additional error to our system, but it is not of importance in this scenario. To give the reader an visualized idea of this error, Figure (5.17) should be given a closer look. A perfect compensation should result in zero position error in the tracking. As seen in the upper error graphs, a small sinusoidal phenomena is

represented due to the numerical error in the derivatives of the reference signal.

Simulation without friction
Figure 5.8 shows the ideal world without friction. The left figure represents the end-effector starting outside the desired trajectory as would be the case in real applications. The right figure is when the end-effector is tracking the circle perfectly without disturbance. The corresponding drive torques are presented in Figure 5.9.

Simulation including Elasto-Plastic friction
Figure 5.10 shows tracking including Elasto-Plastic friction in the manipulator, but without compensation of the friction phenomena. As seen in Figure 5.11 the position errors are evident. The friction torques have also increased in amplitude.

Now we will continue discussing the performance of the different friction models when all parameters are known.

6.2.1 Static Compensation

Figures 5.12 and 5.13 shows the results when using static compensation scheme. We observe the discontinuities when crossing zero velocity. The position tracking error doesn't behave as desired close to zero. At high precision tracking this is very undesirable.

6.2.2 LuGre Compensation

Figures 5.14 and 5.15 shows the results when using dynamic LuGre compensation. The discontinuities in the drive torque is removed and the tracking error is improved. It looks like the error when crossing zero velocity is biased, which agree in comparison to the experimental results in section 5.1 introduced by the Elasto-Plastic friction model.

The LuGre friction model has to include a damping in the $\sigma_1(v)$ parameter to be shown passive. This is a mathematical trick to introduce the passive property of the friction model. By letting $\sigma_1 = 0$ so that the resulting friction force appear as $F = \sigma_1 z + \sigma_2 v$ while still using the $\dot{z}$ dynamics, we reduce the friction model and it can be seen that it becomes passive. This is an approach that is not studied in the literature, and it would be interesting to study this in more detail by doing an experimental comparison of the LuGre model using $\sigma_1 = 0$ and $\sigma_1 \neq 0$.

6.2.3 Elasto-Plastic Compensation

Figures 5.16 and 5.17 shows the results when perfectly tracking is obtained by using the Elasto-Plastic friction model. This is not surprisingly since the friction model used in the

manipulator system is represented by the same model. This comparison is not of much value in a simulation study, thus indeed a compensation scheme based on the Elasto-Plastic model should be performed experimentally.

6.3 Adaptive Friction Compensation

An advanced robotic control system should switch between different control modes depending on the speed, precision, and parameter adaption requirements of the task, [17].

In all practical applications the friction parameters are unknown. Off-line parameter estimation can be time consuming and not always easy accomplished. Suppose a satellite operating in the space needs high precision in servo control to position the antenna pointing towards a specific point miles away. Temperature variations occurring depending on the position of the sun can change the friction parameters significantly. Especially in micro satellites where the weight and dimensions are small, the common way of thermically protecting materials isn't quite as simple as for larger devices. An on-line parameter estimating scheme is needed.

Figure 5.19 shows the reference position and reference velocity. Using the adaption scheme presented in chapter 4.3, we obtain globally asymptotic tracking of the position error e_1 as visualized in Figure 5.20. All parameters are converging, but only β_0 is converging to the true one. The true values are $\beta_0 = 0.5$, $\beta_1 = 0.4$ and $\beta_2 = 0.1$. The stability analysis is not including any proof of the parameters converging to the true ones. They are only bounded. The reason could be that the system is not excited properly, i.e. that the reference signal $q_d(t)$ isn't sufficiently rich. For nonlinear systems this is not quite as simple as for the linear case. In this position tracking scheme, the need for the parameters to converge to the true ones, is not the main issue, whilst the position error is of crucial importance.

This adaptive scheme assumed the stribeck function to be known. An interesting approach using neural networks assumes all parameters including the stribeck function to be unknown.

6.4 Neural Network Approach

Investigation of the neural network approach presented in [9] motivates for a closer look. The friction phenomena were divided into a viscous part, $\theta\dot{x}$ and a second part containing the bristle dynamics, z. By assuming an upper bound of the dynamic friction force and using Lemma 4.4.2, they were able to make the friction dynamics *independent* of the unmeasurable z dynamics, so that it was possible to apply a RBF NN function approximator. Now the need of a bristle state observer is unnecessary.

The main idea of studying this article was to find out if it is possible to apply NN on the

Elasto-Plastic friction model to approximate the unknown $\alpha(\dot{x}, z)$ of the model. Since the smooth function depends on z inside the sinus function, the separation scheme is not that obvious so to create an upper bound. A closer study of NN should be performed in order to verify this. Such a study was not the focus of this thesis, so that we only conclude that a similar approach is not that obvious.

Chapter 7

Conclusions and Recommendations

We have provided an adequate survey of *static* and *dynamic* friction models in a control engineering point of view. A comparing simulation study showed that dynamic friction models are superior to static models when operating at low velocities, and especially when crossing zero velocity. This was also shown using experimental data.

The drift in the LuGre friction model is not that significant without signal noise. The Elasto-Plastic friction model minimize drift, where presliding is elasto-plastic, i.e. under loading the displacement is first purely elastic (reversible) before transitioning to plastic (irreversible).

Adaptive laws were developed for estimating the three unknown parameters in the LuGre friction model. The stribeck curve was assumed to be known. The stability analysis showed that the position tracking error was globally asymptotically stable. All signals were shown to be bounded and it turned out in the simulation study that the estimated parameters didn't converge to the true ones.

An neural network approach were investigated to approximate the dynamic LuGre friction components. It is not straight forward to use the same scheme on the Elasto-Plastic friction model.

7.1 Recommendations for Further Work

- Experimental verification of compensation schemes presented in this report could be performed.
- At Department of Engineering Cybernetics there are available motors at the real time laboratory. Experimental data could be obtained and a more detailed study of the comparison of static vs. dynamic models can be performed without an elastically disturbance. Especially the Elasto-Plastic friction model compared to the LuGre

model would be of interest when noise is exposed to the system. How long is the time interval before the drift is becoming significant?

- Additional experiments should be performed on a comparison of the LuGre model with $\sigma_1 = 0$ and when $\sigma_1 \neq 0$. Does this influence the friction phenomena observed in the LuGre model? What about passivity?
- A closer look to the NN theory could provide knowledge of how to handle the $\alpha(\dot{x}, z)$ function in the Elasto-Plastic friction model.
- Modelling a N-DOF manipulator in interaction with the environment considering friction forces introduced by contact forces would be of interest. Variation in normal forces and friction parameters will give rise to interesting challenges.

Bibliography

[1] B. Armstrong-Hélouvry. *Control of Machines with Friction.* Kluwer Academic Publishers, 1. edition, 1991.

[2] B. Armstrong-Hélouvry, P. Dupont, and C. De Wit. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30:1083–1138, 1994.

[3] V. Augusto. *Differential Models of Hysteresis.* Springer, 1. edition, 1994.

[4] H. Berghuis. *Model-based Robot Control: from Theory to practice.* Enschede, The Netherlands, 1. edition, 1993.

[5] C. Canudas de Wit, H. Olsson, K. Åström, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transaction on Automatic Control*, 40:419–425, 1995.

[6] P. Dupont, B. Armstrong, and V. Hayward. Elasto-plastic friction model: Contact compliance and stiction. *American Control Conference*, 11:1072–1077, 2000.

[7] P. Dupont, V. Hayward, and B. Armstrong. Single state elasto-plastic models. *IEEE Transaction on Automatic Control*, 2001.

[8] O. Egeland and J. Gravdahl. *Modeling and Simulation for Control.* N/A, 1. edition, 2001.

[9] T. Ge, S. S.and Lee and J. Wang. Adaptive nn control of dynamic systems with unknown dynamic friction. *Proceedings of 39th IEEE Conference on Decision and Control*, 2000.

[10] M. Gäfvert. *Comparisons of two Friction Models.* PhD thesis, Lund University, Sweeden, 1996.

[11] M. Gäfvert. Comparisons of two dynamic friction models. *IEEE International Conference on Control Applications*, 20:386–391, 1997.

[12] M. Krasnoselskij and A. Pokrovskij. *Systems with hysteresis.* Springer, New York, 1. edition, 1980.

[13] M. Krstic, I. Kanellakopoulos, and P. Kokotovic. *Nonlinear and Adaptive Control Design.* John Wiley and Sons, inc, 1. edition, 1995.

[14] R. Lozano, B. Brogliato, O. Egeland, and B. Maschke. *Dissipative Systems Analysis and Control.* Springer Verlag, 3. edition, 2000.

[15] H. Nijmeijer and T. Fossen. *Lecture Notes in control and Information Sciences 244.* Springer, 1. edition, 1999.

[16] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators.* Springer, 2. edition, 1999.

[17] J. Slotine and W. Li. Adaptive manipulator control: A case study. *IEEE Transaction on Automatic Control*, 33:995–1003, 1998.

[18] R. Stribeck. Die wesentlichen eigenshaften der gleit- und rollenlager. *Zeitschrift des Vereines Deutscher Ingenieure*, 46:1432–1437, 1902.

[19] K. Åström. Control of systems with friction. 1994.

[20] Y. Tan. *Nonlinear Observer/Controller Design and Its Application to Friction Compensation.* PhD thesis, University of California, Los Angeles, 2000.

[21] A. Telford and J. Moore. Doubly coprime factorizations, reduced-order observers, and dynamic state estimate feedback. *Int. J. Control*, 30:2583–2597, 1989.

Appendix A

Appendix

A.1 Passivity

A physical interpretation of passivity may be made by simply considering a mechanical system (plant and control system) with energy $V(t)$ at time t. The total energy of this system will be the sum of kinetic and potential energy. Let the mechanical energy stored in the system at the initial time $t = t_0$ be denoted by $V(0) \geq 0$. Since the energy $V(T)$ must be positive and lower bounded, it makes sense to define the total system as passive if and only if the mechanical energy dissipated by the system is less than or equal to $V(0)$. Therefore, one of the main consepts of passivity theory is the dissipation of power, [14]. In contrast to the Lyapunov theory, where state variables are considered, passivity theory is based on the input-output properties of a system.

$$V(t) \leq V(0) + \int_0^T y^T(t)u(t)dt \quad [power] \tag{A.1}$$

where the integral represents the external energy inputs. Hence, the rate of change of energy in the system at time t is:

$$\dot{V}(t) \leq y^T(t)u(t) \tag{A.2}$$

where the vector product $y^T u$ simply represents the external power input.

A.2 Properties of L_p signals

In [14] some properties of L_p signals are presented. These properties can be used to analyze the stability of a closed loop control system. Some useful facts are included

Fact 1: If V is a non-decreasing function and if $V \leq M$ for some $M \in R$, then V converges. Example

- $V \geq 0$ and $\dot{V} \leq 0 \Rightarrow V$ converges.

Fact 2 If $f \in L_2$ and $\dot{f} \in L_2$, then $f \to 0$ and $f \in L_\infty$.

Fact 3 If $f_1 \in L_2$ and $f_2 \in L_2$, then $f_1 + f_2 \in L_2$

A.3 Symbols

Table A.1: List of Friction Symbols

Stiffness of surfaces	$\sigma_0 = 0.5$	Nm/rad
Damping coefficient	$\sigma_1 = 0.1$	NMs/rad
Viscous coefficient	$\sigma_2 = 0.3$	Nms/rad
Coulomb friction level	0.285	Nm
Stiction level	$F_s = 0.335$	Nm
Stribeck velocity	$v_s = 0.001$	rad/s

A.4 Experimental comparison

A.4.1 motor.m

```
function [sys,x0] = motor(t,x,u,flag)

if flag == 1,
  %   Konstante parametre
  %
  L_a     = 1.4e-3;
  R_a     = 0.8;
  J_m     = 3.88e-5+3e-4;
  K_T     = 0.056;
  K_E     = K_T;

  F_c     = 0.1;
  F_s     = 1.5;

  w_s   = 0.0001;
  sigma_0  = 200.0;
  sigma_1  = 0.04;
  sigma_2  = 0.045;

  g=(F_c+(F_s-F_c)*exp(-(x(1)/w_s)^2));
  F = sigma_0 * u(1) + sigma_1*(x(1) - sigma_0/g*u(1)*abs(x(1))) + sigma_2*x(1);

  %F = u(1); %For coulomb modell
end

if flag == 1,
  u_a          = u(2);
  T_L          = F;
  %
  i_a          = x(2);
  w_m          = x(1);
  %
  i_a_dot      = (1/L_a)*(-R_a*i_a - K_E*1*w_m + u_a);
  w_m_dot      = (1/J_m)*(K_T*1*i_a - T_L);
  %
  sys(1)       = w_m_dot;
  sys(2)       = i_a_dot;
elseif flag == 0,
```

```
  % return initial conditions
   sys=[2; 0; 2; 2; 0; 0]; % system values [kont,disk,output,input,,]
   x0=[0,0];
elseif flag == 3,
   % return outputs
  sys=[x(1) x(2)]; %output
  else
     sys = [];
end
```

A.4.2 friction.m

```
function [sys,x0] = friction(t,x,u,flag)
% u(1) : velocity

if flag == 1,
  F_c     = 0.1;
  F_s     = 1.5;
  w_s   = 0.0001;
  sigma_0  = 200.0;
  sigma_1  = 0.04;
  sigma_2  = 0.045;

  g=(F_c+(F_s-F_c)*exp(-(u(1)/w_s)^2));
end

if flag == 1,
   % return rates
  sys(1) = u(1) - sigma_0/g*x(1)*abs(u(1));
elseif flag == 0,
  % return initial conditions
   sys=[1; 0; 1; 1; 0; 0]; % system values [kont,disk,output,input,,]
   x0=[0];
elseif flag == 3,
  sys=[x(1)]; %output
  else
     sys = [];
end
```

A.5 Simulation of Static, LuGre and Elasto-Plastic

All simulation studies used the same source code for the `manipulator.m`,`friction.m`,`tracking.m` and `pinit.m`. The static compensation scheme needed no `observer.m`, while the LuGre and the Elasto-Plastic simulation studies had different `observer.m` and `feedback.m`

A.5.1 Common files

manipulator.m

```
function [sys,x0] = manipulator(t,x,u,flag)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Manipulator function
% (C) Ola Jacob Mjøen Iversen olajacob@pvv.ntnu.no
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2-link manipulator from Harry Berghuis (1990) PhD
% x(1)= position link 1 (q1)
% x(2)= velocity link 1 (q1_dot)
% x(3)= position link 2 (q1)
% x(4)= velocity link 2 (q1_dot)
% u(1)= Torque link 1
% u(2)= Torque link 2
% u(3)= z1 bristle state link 1
% u(4)= z2 bristle state link 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global F;
if flag == 1, % Calculating matrices
   %%%%%%%%%%
   % Friction on/off
   on = 1;
   off = 0;
   val = on;
   %%%%%%%%%%
   rad = pi/180; %til radianer


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Manipulator properties
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   q1 = x(1);
   q1_dot = x(2);
```

```
   q2 = x(3);
   q2_dot = x(4);

   d11 = 1.02 * cos(q2*rad) + 8.77; %        [ d11 d12 ]
   d12 = 0.76 + 0.51 * cos(q2*rad); % D(q) = [         ]
   d21 = 0.76 + 0.51 * cos(q2*rad); %        [ d21 d22 ]
   d22 = 0.62;

   dc11 = ( 1.02 * cos(x(3)*rad)+8.77 ) * (-0.51 * sin(x(3)*rad) * x(4))
      + (0.76+0.51 * cos(x(3)*rad)) * (0.51 * sin(x(3)*rad) * x(2) );
   dc12 = ( 1.02 * cos(x(3)*rad)+8.77 ) * ( -0.51 * sin(x(3)*rad)*(x(2) + x(4)) );
   dc21 = ( 0.76 + 0.51 * cos(x(3)*rad) ) * (-0.51 * sin(x(3)*rad) * x(4))
      + ( 0.62 * 0.51 * sin(x(3)*rad) * x(2) );
   dc22 = ( 0.76 + 0.51 * cos(x(3)*rad) ) * (-0.51 * sin(x(3)*rad) * (x(2) + x(4)) );

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Friction dynamics. Elasto-Plastic
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 pinit;

   z1 = u(3);
   z2 = u(4);

   s1=(F_c+(F_s-F_c)*exp(-(x(2)/w_s)^2));
   s2=(F_c+(F_s-F_c)*exp(-(x(4)/w_s)^2));

   if sign(x(2)) == sign(z1),

       if abs(z1) < z_ba,
     a1=0;
    elseif abs(z1) >= z_ba & abs(z1) < z_max,
     a1=(1/2)*sin(pi*(z1-0.5*(z_max+z_ba))/(z_max-z_ba))+0.5;
     elseif abs(z1) >= z_max,
       a1=1;
     end

   else
        a1 = 0;
   end

   if sign(x(4)) == sign(z2),

         if abs(z2) < z_ba,
```

```
          a2=0;
         elseif abs(z2) >= z_ba & abs(z2) < z_max,
          a2=(1/2)*sin(pi*(z2-0.5*(z_max+z_ba))/(z_max-z_ba))+0.5;
         elseif abs(z2) >= z_max,
          a2=1;
         end

   else
         a2 = 0;
   end

   h1 = sigma_0*a1/s1;
   h2 = sigma_0*a2/s2;

   F_1 = (sigma_0 * z1 + sigma_1*(x(2)*(1 - h1*z1*sign(x(2)) )) + sigma_2*x(2))*val;
   F_2 = (sigma_0 * z2 + sigma_1*(x(4)*(1 - h2*z2*sign(x(4)) )) + sigma_2*x(4))*val;

F=F_1; % used for error detection

   tau_1 = u(1) - F_1;
   tau_2 = u(2) - F_2;
end


if flag == 1,
    % return states
 sys(1) = x(2);
 sys(2) = d11*tau_1+d12*tau_2-(dc11*x(2)+dc12*x(4));
 sys(3) = x(4);
 sys(4) = d21*tau_1+d22*tau_2-(dc21*x(2)+dc22*x(4));

elseif flag == 0,

  % return initial conditions
   sys=[4; 0; 5; 4; 0; 0]; % system values
   v = tracking(0);
   x0=[v(3);v(5);v(4);v(7)]; %Initial betingelser, kalkulert fra tracking(0)
  x0=[1.55;0;-1.75;0];
elseif flag == 3,
   % return outputs
  sys=[x(1) x(2) x(3) x(4) F]; %output
else
     sys = [];
```

```
end
```

friksjon.m

```
function [sys,x0] = friksjon(t,x,u,flag)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Elasto-Plastic friction model
% (C) Ola Jacob Mjøen Iversen olajacob@pvv.ntnu.no
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Outputs:
%  x(1)= Friction internal bristle state link 1
%  x(2)= Friction internal bristle state link 2
% Inputs:
%  u(1)= velocity link 1 (q1_dot)
%  u(2)= velocity link 2 (q2_dot)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if flag == 1, % Friction values and Stribeck function

 pinit;

     z1 = x(1);
     z2 = x(2);

     g1=(F_c+(F_s-F_c)*exp(-(u(1)/w_s)^2));
     g2=(F_c+(F_s-F_c)*exp(-(u(2)/w_s)^2));

     if sign(u(1)) == sign(z1),

      if abs(z1) < z_ba,
    a1=0;
    elseif abs(z1) >= z_ba & abs(z1) < z_max,
    a1=(1/2)*sin(pi*(z1-0.5*(z_max+z_ba))/(z_max-z_ba))+0.5;
    elseif abs(z1) >= z_max,
    a1=1;
    end

     else
      a1 = 0;
     end
```

```
    if sign(u(2)) == sign(z2),

        if abs(z2) < z_ba,
        a2=0;
        elseif abs(z2) >= z_ba & abs(z2) < z_max,
           a2=(1/2)*sin(pi*(z2-0.5*(z_max+z_ba))/(z_max-z_ba))+0.5;
        elseif abs(z2) >= z_max,
        a2=1;
        end
     else
         a2 = 0;
     end

  h1 = sigma_0*a1/g1;
  h2 = sigma_0*a2/g2;

end

if flag == 1, % another "if" for the sake of clairity
   % return rates
   sys(1) = u(1)*(1 - h1*x(1)*sign(u(1)) );
   sys(2) = u(2)*(1 - h2*x(2)*sign(u(2)) );

elseif flag == 0,

  % return initial conditions
   sys=[2; 0; 2; 2; 0; 0]; % system values [kont,disk,output,input,,]


   x0=[0;0]; %Initial betingelser,

elseif flag == 3,
   % return outputs
  sys=[x(1) x(2)]; %output [Z1,Z2]

  else
     sys = [];
end
```

tracking.m

```
function v = tracking(t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Tracking circle
% (C) Ola Jacob Mjøen Iversen olajacob@pvv.ntnu.no
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Outputs: v = [v1,v2,vd1,vdd1,vd2,vdd2]
%      v1  : desired angle position link 1
%      v2  : desired angle position link 2
%      vd1 : desired angle velocity link 1
%      vdd1: desired angle acceleration link 1
%      vd2 : desired angle velocity link 2
%      vdd2: desired angle acceleration link 2
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% a1 = a2 = 1 [m]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Constants
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 0.4; % Radius på referansesirkelen
ox = 1; % Sentrum (ox,oy) av sirkelen
oy = 0.8;
w=0.7; % angular frequency

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculating reference circle
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
px = ox + r*cos(w*t);
py = oy + r*sin(w*t);

px_dot = -0.1*r*sin(w*t);
py_dot = 0.1*r*cos(w*t);

px_ddot = -0.01*r*cos(w*t);
py_ddot = -0.01*r*sin(w*t);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Inverse kinematics
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cq2 = (px^2+py^2-2)/2;
```

```
sq2 = -sqrt(1-cq2^2);

v2 = atan2(sq2,cq2);

sq1 = ((1+cq2)*py-sq2*px)/(px^2+py^2);
cq1 = ((1+cq2)*px+sq2*py)/(px^2+py^2);

v1 = atan2(sq1,cq1);

a=0.0355;

vd1=5e-3+a*sin(w*t);
vdd1=w*a*cos(w*t);

vd2=-(-(-(ox+r*cos(w*t))*r*sin(w*t)*w+(oy+r*sin(w*t))*r*cos(w*t)*w)
/(sqrt(1-(1/2*(ox+r*cos(w*t))^2
+1/2*(oy+r*sin(w*t))^2-1)^2))-sqrt(1-(1/2*(ox+r*cos(w*t))^2
+1/2*(oy+r*sin(w*t))^2-1)^2)*(-(ox+r*cos(w*t))*r*sin(w*t)*w
+(oy+r*sin(w*t))*r*cos(w*t)*w)/((1/2*(ox+r*cos(w*t))^2+1/2*(oy+r*sin(w*t))^2-1)^2))
/(1+(1-(1/2*(ox+r*cos(w*t))^2+1/2*(oy+r*sin(w*t))^2-1)^2)/
((1/2*(ox+r*cos(w*t))^2+1/2*(oy+r*sin(w*t))^2-1)^2));

vdd2=-(-(-(ox+r*cos(w*t))*r*sin(w*t)*w+(oy+r*sin(w*t))*r*cos(w*t)*w)^2
*(1/2*(ox+r*cos(w*t))^2
+1/2*(oy+r*sin(w*t))^2-1)/((1-(1/2*(ox+r*cos(w*t))^2+1/2
*(oy+r*sin(w*t))^2-1)^2)^(3/2))
-(r^2*sin(w*t)^2*w^2-(ox+r*cos(w*t))*r*cos(w*t)*w^2+r^2*cos(w*t)^2
*w^2-(oy+r*sin(w*t))*r*sin(w*t)*w^2)
/(sqrt(1-(1/2*(ox+r*cos(w*t))^2+1/2*(oy+r*sin(w*t))^2-1)^2))+(-(ox+r*cos(w*t))
*r*sin(w*t)*w
+(oy+r*sin(w*t))*r*cos(w*t)*w)^2/(sqrt(1-(1/2*(ox+r*cos(w*t))^2
+1/2*(oy+r*sin(w*t))^2-1)^2)
*(1/2*(ox+r*cos(w*t))^2+1/2*(oy+r*sin(w*t))^2-1))
+2*sqrt(1-(1/2*(ox+r*cos(w*t))^2
+1/2*(oy+r*sin(w*t))^2-1)^2)*(-(ox+r*cos(w*t))*r*sin(w*t)*w+(oy+r*sin(w*t))
*r*cos(w*t)*w)^2
/((1/2*(ox+r*cos(w*t))^2+1/2*(oy+r*sin(w*t))^2-1)^3)
-sqrt(1-(1/2*(ox+r*cos(w*t))^2
+1/2*(oy+r*sin(w*t))^2-1)^2)*(r^2*sin(w*t)^2*w^2-(ox+r*cos(w*t))
*r*cos(w*t)*w^2
+r^2*cos(w*t)^2*w^2-(oy+r*sin(w*t))*r*sin(w*t)*w^2)/((1/2*(ox+r*cos(w*t))^2
+1/2*(oy+r*sin(w*t))^2-1)^2))/(1+(1-(1/2*(ox+r*cos(w*t))^2
+1/2*(oy+r*sin(w*t))^2-1)^2)
```

```
/((1/2*(ox+r*cos(w*t))^2+1/2*(oy+r*sin(w*t))^2-1)^2))
+(-(-(ox+r*cos(w*t))*r*sin(w*t)*w
+(oy+r*sin(w*t))*r*cos(w*t)*w)/(sqrt(1-(1/2*(ox+r*cos(w*t))^2
+1/2*(oy+r*sin(w*t))^2-1)^2))
-sqrt(1-(1/2*(ox+r*cos(w*t))^2+1/2*(oy+r*sin(w*t))^2-1)^2)*(-(ox+r*cos(w*t))
*r*sin(w*t)*w
+(oy+r*sin(w*t))*r*cos(w*t)*w)/((1/2*(ox+r*cos(w*t))^2
+1/2*(oy+r*sin(w*t))^2-1)^2))
*(-2*(-(ox+r*cos(w*t))*r*sin(w*t)*w+(oy+r*sin(w*t))*r*cos(w*t)*w)
/(1/2*(ox+r*cos(w*t))^2
+1/2*(oy+r*sin(w*t))^2-1)-2*(1-(1/2*(ox+r*cos(w*t))^2+1/2*(oy+r*sin(w*t))^2-1)^2)
*(-(ox+r*cos(w*t))
*r*sin(w*t)*w+(oy+r*sin(w*t))*r*cos(w*t)*w)/((1/2*(ox+r*cos(w*t))^2
+1/2*(oy+r*sin(w*t))^2-1)^3))
/((1+(1-(1/2*(ox+r*cos(w*t))^2+1/2*(oy+r*sin(w*t))^2-1)^2)
/((1/2*(ox+r*cos(w*t))^2+1
/2*(oy+r*sin(w*t))^2-1)^2))^2);

vv = [v1,v2,vd1,vdd1,vd2,vdd2];

v = [v1,v2,vv];
```

pinit.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% pinit.m
% Initialisation of friction parameters etc.
%
% mars 2002, Ola Jacob Mjøen Iversen
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sigma_0 = 10;  % [Nm/rad] Stiffness of surfaces
sigma_1 = 5.4; % [Nm/rad] Damping coefficient
sigma_2 = 0.5; % [Nms/rad] Viscous coefficient
F_c = 1;       % [Nm] Coulomb friction level
F_s = 1.1;     % [Nm] Stiction level
w_s = 0.1;     % [rad/s] Stribeck velocity

z_max = 0.1;   % [m]
```

```
z_ba = 0.0008; % [m]
```

A.5.2 Static Friction Compensation

feedback.m

```
\include{feedback_static}
```

A.5.3 LuGre Friction Compensation

observer.m

```
function [sys,x0] = observer(t,x,u,flag)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bristle state Observer
% (C) Ola Jacob Mjøen Iversen olajacob@pvv.ntnu.no
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Outputs:
%  x(1)= Friction internal bristle state link 1
%  x(2)= Friction internal bristle state link 2
% Inputs:
%  u(1)= velocity link 1 (q1_dot)
%  u(2)= velocity link 2 (q2_dot)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if flag == 1, % Friction values and Stribeck function
pinit;

     g1=(F_c+(F_s-F_c)*exp(-(u(1)/w_s)^2));
     g2=(F_c+(F_s-F_c)*exp(-(u(2)/w_s)^2));
end

if flag == 1, % another "if" for the sake of clairity
   % return rates
   sys(1) = u(1) - sigma_0*x(1)*abs(u(1))/g1;
   sys(2) = u(2) - sigma_0*x(2)*abs(u(2))/g2;
```

```
elseif flag == 0,

  % return initial conditions
   sys=[2; 0; 2; 2; 0; 0]; % system values [kont,disk,output,input,,]


   x0=[0;0]; %Initial betingelser,

elseif flag == 3,
   % return outputs
  sys=[x(1) x(2)]; %output [Z1,Z2]

  else
     sys = [];
end
```

feedback.m

```
function tau_c = feedback(x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Feedback linearization control
% (C) Ola Jacob Mjøen Iversen olajacob@pvv.ntnu.no
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Outputs: tau_c = [tau1,tau2]
%      tau1: control torque link 1
%      tau2: control torque link 2
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Inputs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
q1 = x(1);
q2 = x(2);
qm1_dot = x(3);  % "measured" velocity link 1
qm2_dot = x(4);  % "measured" velocity link 2

q1_dot = x(3);
q2_dot = x(4);
```

```
qd1_dot = x(7);
qd2_dot = x(9);
qd1_dot_dot = x(8);
qd2_dot_dot = x(10);

z1 = x(11);
z2 = x(12);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Constants
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pinit;
rad = pi/180; %til radianer

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Manipulator properties
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
d11 = 1.02 * cos(q2*rad) + 8.77;
d12 = 0.76 + 0.51 * cos(q2*rad);
d21 = 0.76 + 0.51 * cos(q2*rad);
d22 = 0.62;

c11 = -0.51 * sin(q2*rad)*q2_dot;
c12 = -0.51 * sin(q2*rad)*(q1_dot + q2_dot);
c21 = 0.51 * sin(q2*rad)*q1_dot;
c22 = 0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Feedback linearisation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tau_1 = d11 * qd1_dot_dot + d12 * qd2_dot_dot + c11 * qd1_dot + c12 * qd2_dot;
tau_2 = d21 * qd1_dot_dot + d22 * qd2_dot_dot + c21 * qd1_dot + c22 * qd2_dot;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LuGre Friction Compensation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

g1=(F_c+(F_s-F_c)*exp(-(q1_dot/w_s)^2));
g2=(F_c+(F_s-F_c)*exp(-(q2_dot/w_s)^2));

z1_prikk = q1_dot - sigma_0*z1*abs(q1_dot)/g1;
z2_prikk = q2_dot - sigma_0*z2*abs(q2_dot)/g2;
```

```
tauf1 = sigma_0*z1 + sigma_1*z1_prikk + sigma_2*q1_dot;
tauf2 = sigma_0*z2 + sigma_1*z2_prikk + sigma_2*q2_dot;

tau1=tau_1+tauf1;
tau2=tau_2+tauf2;

tau_c = [tau1;tau2;tauf1]; % Control Outputs
```

A.5.4 Elasto-Plastic Friction Compensation

observer.m

```
function [sys,x0] = observer(t,x,u,flag)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dual-Bristle state Observer
% (C) Ola Jacob Mjøen Iversen olajacob@pvv.ntnu.no
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Outputs:
%  x(1)= Friction internal bristle state link 1
%  x(2)= Friction internal bristle state link 2
% Inputs:
%  u(1)= velocity link 1 (q1_dot)
%  u(2)= velocity link 2 (q2_dot)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if flag == 1, % Friction values and Stribeck function

     pinit; % Set the friction Parameters

   z1 = x(1);
   z2 = x(2);

   s1=(F_c+(F_s-F_c)*exp(-(u(1)/w_s)^2));
   s2=(F_c+(F_s-F_c)*exp(-(u(2)/w_s)^2));

   if sign(u(1)) == sign(z1),

       if abs(z1) < z_ba,
```

```
     a1=0;
    elseif abs(z1) >= z_ba & abs(z1) < z_max,
     a1=(1/2)*sin(pi*(z1-0.5*(z_max+z_ba))/(z_max-z_ba))+0.5;
     elseif abs(z1) >= z_max,
       a1=1;
     end

   else
        a1 = 0;
   end

   if sign(u(2)) == sign(z2),

         if abs(z2) < z_ba,
          a2=0;
         elseif abs(z2) >= z_ba & abs(z2) < z_max,
          a2=(1/2)*sin(pi*(z2-0.5*(z_max+z_ba))/(z_max-z_ba))+0.5;
         elseif abs(z2) >= z_max,
          a2=1;
         end

   else
         a2 = 0;
   end

   h1 = sigma_0*a1/s1;
   h2 = sigma_0*a2/s2;

   F_1 = sigma_0 * z1 + sigma_1*(u(1)*(1 - h1*z1*sign(u(1)) )) + sigma_2*u(1);
   F_2 = sigma_0 * z2 + sigma_1*(u(2)*(1 - h2*z2*sign(u(2)) )) + sigma_2*u(2);
end

if flag == 1, % another "if" for the sake of clairity
   % return rates
   sys(1) = u(1)*(1 - h1*x(1)*sign(u(1)) );
   sys(2) = u(2)*(1 - h2*x(2)*sign(u(2)) );

elseif flag == 0,

  % return initial conditions
   sys=[2; 0; 2; 2; 0; 0]; % system values [kont,disk,output,input,,]
```

```
   x0=[0;0]; %Initial betingelser,

elseif flag == 3,
   % return outputs
  sys=[x(1) x(2)]; %output [Z1,Z2]

  else
     sys = [];
end
```

feedback.m

```
function tau_c = feedback(x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Feedback linearization control
% (C) Ola Jacob Mjøen Iversen olajacob@pvv.ntnu.no
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Outputs: tau_c = [tau1,tau2]
%      tau1: control torque link 1
%      tau2: control torque link 2
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Inputs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
q1 = x(1);
q2 = x(2);
qm1_dot = x(3);  % "measured" velocity link 1
qm2_dot = x(4);  % "measured" velocity link 2

q1_dot = x(3);
q2_dot = x(4);

qd1_dot = x(7);
qd2_dot = x(9);
qd1_dot_dot = x(8);
qd2_dot_dot = x(10);

z1 = x(11);
```

```
z2 = x(12);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Constants
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pinit;

rad = pi/180; %til radianer

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Manipulator properties
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
d11 = 1.02 * cos(q2*rad) + 8.77;
d12 = 0.76 + 0.51 * cos(q2*rad);
d21 = 0.76 + 0.51 * cos(q2*rad);
d22 = 0.62;

c11 = -0.51 * sin(q2*rad)*q2_dot;
c12 = -0.51 * sin(q2*rad)*(q1_dot + q2_dot);
c21 = 0.51 * sin(q2*rad)*q1_dot;
c22 = 0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Feedback linearisation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tau_1 = d11 * qd1_dot_dot + d12 * qd2_dot_dot + c11 * qd1_dot + c12 * qd2_dot;
tau_2 = d21 * qd1_dot_dot + d22 * qd2_dot_dot + c21 * qd1_dot + c22 * qd2_dot;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Elasto-Plastic Friction Compensation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s1=(F_c+(F_s-F_c)*exp(-(q1_dot/w_s)^2));
s2=(F_c+(F_s-F_c)*exp(-(q2_dot/w_s)^2));

   if sign(q1_dot) == sign(z1),

       if abs(z1) < z_ba,
     a1=0;
    elseif abs(z1) >= z_ba & abs(z1) < z_max,
     a1=(1/2)*sin(pi*(z1-0.5*(z_max+z_ba))/(z_max-z_ba))+0.5;
     elseif abs(z1) >= z_max,
       a1=1;
```

```
     end

   else
        a1 = 0;
   end

   if sign(q2_dot) == sign(z2),

         if abs(z2) < z_ba,
          a2=0;
         elseif abs(z2) >= z_ba & abs(z2) < z_max,
          a2=(1/2)*sin(pi*(z2-0.5*(z_max+z_ba))/(z_max-z_ba))+0.5;
         elseif abs(z2) >= z_max,
          a2=1;
         end

   else
         a2 = 0;
   end

h1 = sigma_0*a1/s1;
h2 = sigma_0*a2/s2;

tauf1 = sigma_0 * z1 + sigma_1*(q1_dot*(1 - h1*z1*sign(q1_dot) )) + sigma_2*q1_dot;
tauf2 = sigma_0 * z2 + sigma_1*(q2_dot*(1 - h2*z2*sign(q2_dot) )) + sigma_2*q2_dot;

tau1=tau_1+tauf1;
tau2=tau_2+tauf2;

tau_c = [tau1;tau2;tauf1]; % Control Outputs, tauf1 for measurement purpose
```

A.6 Adaptive Friction Compensation

A.6.1 manipulator.m

```
function v_dot = manipulator(u)

q1 = u(1);
q2 = u(2);
```

```
q1_dot = u(3);
q2_dot = u(4);

tau = [u(5);u(6)];
z1 = u(7);
z2 = u(8);

z = [z1;z2];

S0 = diag([0.5,0.5]);
S1 = diag([0.1,0.1]);
S2 = diag([0.3,0.3]);

% Manipulator constants
l1=1;
l2=1;
m1=5;
m2=5;


% Christoffel symbols, c_ijk
c111 = 0;
c222 = 0;
c112 = m2*l1*l2/2*sin(q2);
c221 = -m2*l1*l2/2*sin(q2);
c121 = -m2*l1*l2/2*sin(q2);
c211 = c121;
c122 = 0;
c212 = c122;

M = [m1*l1^2/3+m2*(l1+l2^2/2+l1*l1*cos(q2)), m2*(l2^2/3+l1*l2/2*cos(q2))
   m2*(l2^2/3+l1*l2*cos(q2)/2), m2*l2^2/3];

C = [c111*q1_dot + c211*q2_dot, c121*q1_dot + c221*q2_dot
   c112*q1_dot + c212*q2_dot, c122*q1_dot + c222*q2_dot];

Minv = M^-1;

v = [q1_dot;q2_dot];

Fs = 0.335;
Fc = 0.285;
vs = 0.01;
```

```
g1 = Fc+(Fs-Fc)*exp(-(q1_dot/vs)^2);
g2 = Fc+(Fs-Fc)*exp(-(q2_dot/vs)^2);

G=diag([1/g1,1/g2]);
F = S0*z + S1*(v - triu(triu(G*(abs(v)*z'))')*[1;1]) + S2*v;
%F=[0;0];

v_dot = Minv*tau - Minv*F - Minv*C*v;
```

A.6.2 observer.m

```
function [sys,x0] = observer(t,x,u,flag)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dual bristle state Observer
% (C) Ola Jacob Mjøen Iversen olajacob@pvv.ntnu.no
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Outputs:
%   x(1) : hat_z_01
%  x(2) : hat_z_11
%  x(3) : hat_beta_01
%  x(4) : hat_beta_11
%  x(5) : hat_beta_21
%   x(6) : hat_z_02
%  x(7) : hat_z_12
%  x(8) : hat_beta_02
%  x(9) : hat_beta_12
%  x(10): hat_beta_22
% Inputs:
%  u(1) : q1
%  u(2) : q2
%  u(3) : q1_dot
%  u(4) : q2_dot
%  u(5) : q1_ref
%  u(6) : q2_ref
%  u(7) : q1_dot_ref
%  u(8) : q2_dot_ref
%  u(9) : q1_ddot_ref
%  u(10): q2_ddot_ref
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
if flag == 1, % Friction values and Stribeck function

  pinit;
  q1 = u(1);
  q2 = u(2);
  q1_dot = u(3);
  q2_dot = u(4);
  q1_ref = u(5);
  q2_ref = u(6);
  q1_dot_ref = u(7);
  q2_dot_ref = u(8);

  F_c = 0.285;   % [Nm] Coulomb friction level
  F_s = 0.335;   % [Nm] Stiction level
  w_s = 0.01;    % [rad/s] Stribeck velocity

  gamma_01 = 10;
  gamma_11 = 20;
  gamma_21 = 6;
  gamma_02 = 10;
  gamma_12 = 20;
  gamma_22 = 6;

  g =(F_c+(F_s-F_c)*exp(-(q1_dot/w_s)^2));
  g2=(F_c+(F_s-F_c)*exp(-(q2_dot/w_s)^2));

  e1_1 = q1-q1_ref;
  e1_2 = q2-q2_ref;

  e2_1 = q1_dot - q1_dot_ref + k11*e1_1;
  e2_2 = q2_dot - q2_dot_ref + k12*e1_2;

end

if flag == 1, % another "if" for the sake of clairity
 if abs(e1_1)>1.5*10^(-8),
% Dual-Observer joint 1
  sys(1) = q1_dot - abs(q1_dot)*x(1)/g-e2_1;
  sys(2) = q1_dot - abs(q1_dot)*x(2)/g + abs(q1_dot)*e2_1/g;
% Adaptive laws joint 1
```

```
  sys(3) = -gamma_01 * e2_1*x(1);
  sys(4) = -gamma_11 * q1_dot*e2_1;
  sys(5) = gamma_21 * abs(q1_dot)*x(2)/g*e2_1;
 else
  sys(1) = 0;
  sys(2) = 0;
  sys(3) = 0;
  sys(4) = 0;
  sys(5) = 0;
 end
 if abs(e1_2)>1.5*10^(-8),
% Dual-Observer joint 2
  sys(6) = q2_dot - abs(q2_dot)*x(6)/g2-e2_2;
  sys(7) = q2_dot - abs(q2_dot)*x(7)/g2 + abs(q2_dot)*e2_2/g2;
% Adaptive laws joint 2
  sys(8) = -gamma_02 * e2_2*x(6);
  sys(9) = -gamma_12 * q2_dot*e2_2;
  sys(10) = gamma_22 * abs(q2_dot)*x(7)/g2*e2_2;
 else
  sys(6) = 0;
  sys(7) = 0;
  sys(8) = 0;
  sys(9) = 0;
  sys(10) = 0;
 end

elseif flag == 0,

   % return initial conditions
    sys=[10; 0; 10; 10; 0; 0]; % system values [kont,disk,output,input,,]

    x0=[0,0,0,0,0,0,0,0,0,0];

 elseif flag == 3,
    % return outputs

   sys=[x(1) x(2) x(3) x(4) x(5) x(6) x(7) x(8) x(9) x(10)]; %output
   else
      sys = [];
end
```

A.6.3 friction.m

```
function z_dot=friction(x)

v1 = x(1);
v2 = x(2);
z1 = x(3);
z2 = x(4);

v = [v1;v2];
z = [z1;z2];

Fs = 0.335;
Fc = 0.285;
vs = 0.01;

g1 = Fc+(Fs-Fc)*exp(-(v1/vs)^2);
g2 = Fc+(Fs-Fc)*exp(-(v2/vs)^2);

G=diag([1/g1,1/g2]);

z_dot = v - triu(triu(G*(abs(v)*z'))')*[1;1];
```

A.6.4 controller.m

```
function ut = controller(x)
%-------------------------------------------------------------------
% Inputs
%-------------------------------------------------------------------

q1  = x(1);
q2  = x(2);
v1  = x(3);
v2  = x(4);
qd1 = x(5);
qd2 = x(6);
vd1 = x(7);
vd2 = x(8);
vdd1= x(9);
vdd2= x(10);

z01 = x(11);
```

```
z11 = x(12);
beta_01 = x(13);
beta_11 = x(14);
beta_21 = x(15);
z02 = x(16);
z12 = x(17);
beta_02 = x(18);
beta_12 = x(19);
beta_22 = x(20);

q  = [q1;q2];
v  = [v1;v2];
qd = [qd1;qd2];
vd = [vd1;vd2];
vdd= [vdd1;vdd2];

e     = q - qd;
e_dot = v - vd;

%-------------------------------------------------------------------------
% Manipulator constants
%-------------------------------------------------------------------------
l1=1;
l2=1;
m1=5;
m2=5;

pinit;
%-------------------------------------------------------------------------
% Christoffel symbols, c_ijk
%-------------------------------------------------------------------------
c111 = 0;
c222 = 0;
c112 = m2*l1*l2/2*sin(q2);
c221 = -m2*l1*l2/2*sin(q2);
c121 = -m2*l1*l2/2*sin(q2);
c211 = c121;
c122 = 0;
c212 = c122;

lambda1 = 5;
lambda2 = 5;
Kp = diag([lambda1^2,lambda2^2]);
```

```
Kd = diag([2*lambda1,2*lambda2]);
A=diag([5,5]);

alfa = vd - A*e;

s = v - alfa;

alfa_dot = vdd - A*(v-vd);

M = [m1*l1^2/3+m2*(l1+l2^2/2+l1*l1*cos(q2)), m2*(l2^2/3+l1*l2/2*cos(q2))
   m2*(l2^2/3+l1*l2*cos(q2)/2), m2*l2^2/3];

C = [c111*v1 + c211*v2, c121*v1 + c221*v2
   c112*v1 + c212*v2, c122*v1 + c222*v2];

g1=(F_c+(F_s-F_c)*exp(-(v1/w_s)^2));
g2=(F_c+(F_s-F_c)*exp(-(v2/w_s)^2));

F_hat_1 = beta_01*z01 + beta_11*v1 - beta_21*abs(v1)*z11/g1;
F_hat_2 = beta_02*z02 + beta_12*v2 - beta_22*abs(v2)*z12/g2;

F=[F_hat_1;F_hat_2];

u = M*alfa_dot + C*alfa - Kd*s - Kp*e + F;

ut=[u;e;e_dot;s];
```

A.6.5 trajectory.m

```
function d = trajectory(t)

w=0.1 % angular frequency

qd1 = pi*(1-cos(w*t))/3;
qd2 = pi*(1-cos(w*t))/3;

vd1 = pi*w*sin(w*t)/3;
vd2 = pi*w*sin(w*t)/3;

vdd1 = pi*w^2*cos(w*t)/3;
vdd2 = pi*w^2*cos(w*t)/3;
```

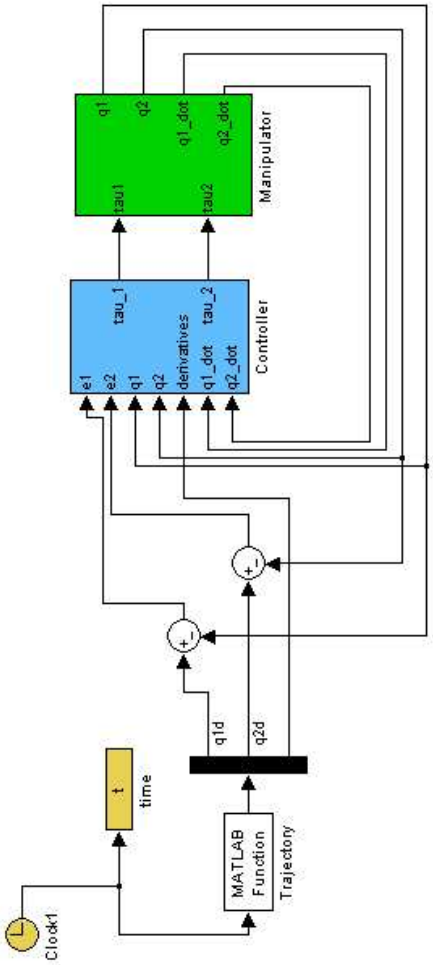
```
d = [qd1 qd2 vd1 vd2 vdd1 vdd2];
```

A.6.6 pinit.m

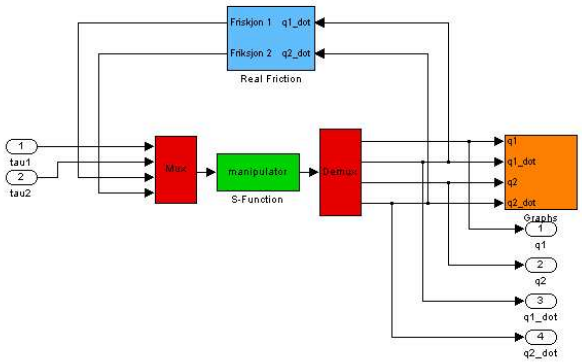
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% pinit.m
% Initialisation of friction parameters etc.
%
% (C) 2002, Ola Jacob Mjøen Iversen
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sigma_0 = 0.5; % [Nm/rad] Stiffness of surfaces
sigma_1 = 0.1; % [Nm/rad] Damping coefficient
sigma_2 = 0.3; % [Nms/rad] Viscous coefficient
F_c = 0.285;   % [Nm] Coulomb friction level
F_s = 0.335;   % [Nm] Stiction level
w_s = 0.01;    % [rad/s] Stribeck velocity

lambda1 = 5;
lambda2 = 5;
Kp = diag([lambda1^2,lambda2^2]);
Kd = diag([2*lambda1,2*lambda2]);
A=diag([5,5]);

k11=A(1);
k12=A(4);

k21=Kd(1);
k22=Kd(4);
```

A.7 Simulink

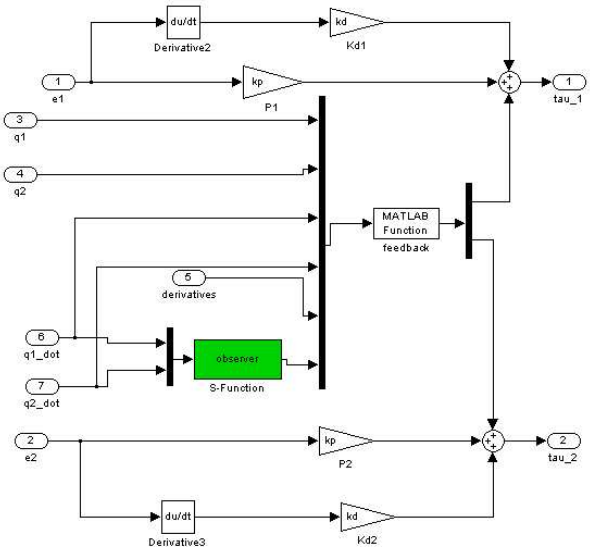
A.7.1 Simulink System with known Parameters



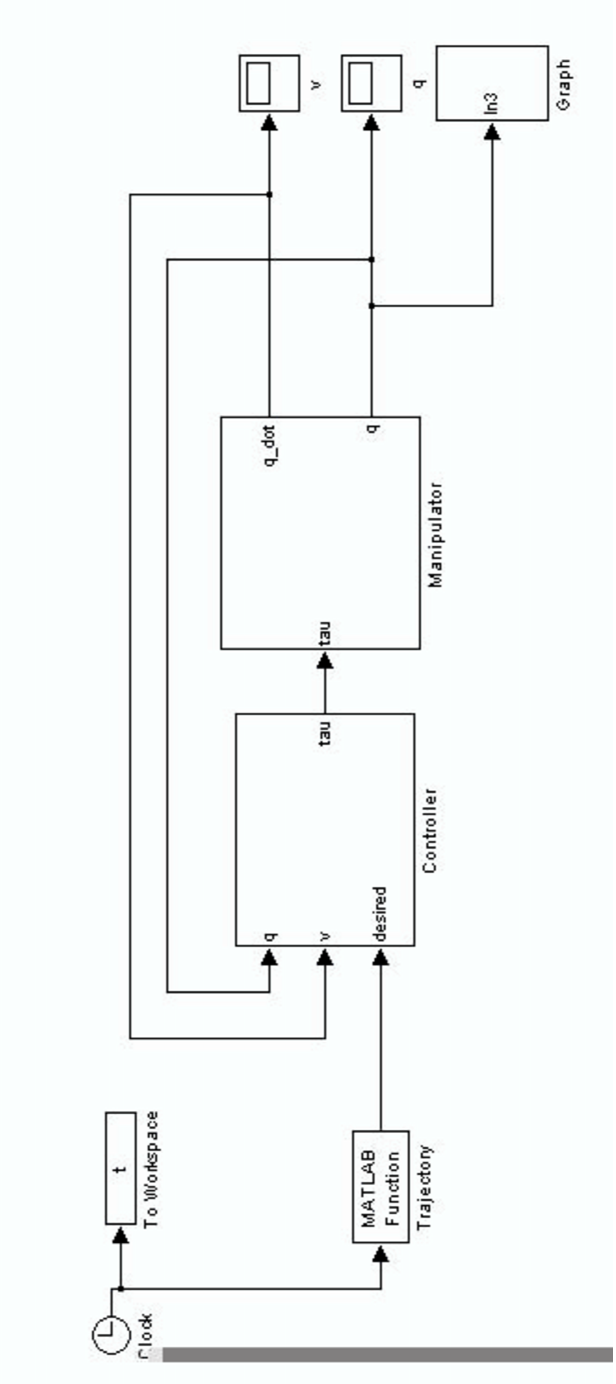
A.7.2 Inside Manipulator



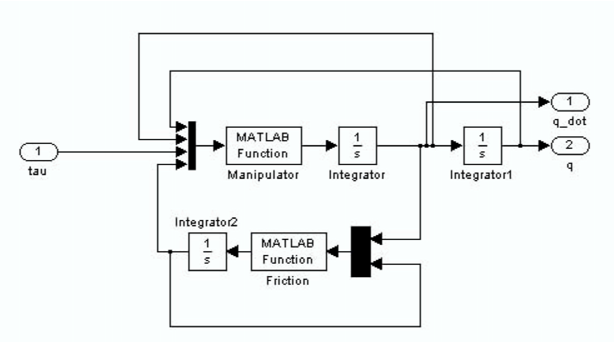
A.7.3 Inside Controller



A.7.4 Simulink System of Adaptive Scheme



A.7.5 Inside Manipulator



A.7.6 Inside Controller