NTNU
Innovation and Creativity

Discrete-Time Backstepping Design Applied to Position Tracking Control of an Electro-Pneumatic Clutch Actuator

Trond Willi Isaksen

Master of Science in Engineering Cybernetics
Submission date: June 2007
Supervisor: Tor Arne Johansen, ITK
Co-supervisor: Håvard Fjær Grip, ITK
Christian Bratli, Kongsberg Automotive
Stein Roar Snare, Kongsberg Automotive

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

Backstepping has become a popular strategy for industrial nonlinear control systems. In this project, backstepping is used to control an electro-pneumatic clutch actuator for use in heavy duty trucks. Because of the low sample rates often found in automotive industry, we would like to learn more about differences between discrete and continuous backstepping designs. During this project, the student will design, test and compare controllers based on continuous and discrete backstepping.

Some topics that will be adressed are:
- Literature study on available theory for discrete-time backstepping.
- Full-state feedback backstepping in continuous time.
- Full-state feedback backstepping in discrete time.

Assignment given: 15. January 2007
Supervisor: Tor Arne Johansen, ITK

Preface

The work in this master thesis was carried out at the Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU) in Trondheim during the spring 2007. The thesis has been written on behalf of Kongsberg Automotive AS (KA) on the subject of discrete-time backstepping control of nonlinear sampled-data systems.

The thesis is both a study of available literature on the topic, as well as theory extracted to use on KA's electro-pneumatic clutch actuator. The thesis is based on work by Kaasa (2003) and Løkken (2006), as well as my own project work last autumn.

Having adopted a slightly informal and personal style of writing means that I refer to myself and the reader whenever the personal pronoun ”we” is used, since the work in this thesis is entirely my own.

After five long years of studying I am very pleased to graduate from NTNU, and would therefore like to take the opportunity to thank NTNU for my time at the university.

Trondheim, June 12, 2007

Trond-Willi Isaksen

Abstract

This thesis investigates different methods of backstepping controller design for an electro-pneumatic clutch actuator used in heavy duty trucks. The first part of the thesis is a literature study, where the subject is control of nonlinear-sampled data systems in general. Sampled-data systems contain a continuous-time plant and a digitally implemented controller, which in general make them harder to analyze and control than systems that operate purely in the continuous-time or discrete-time domain. The available theory of nonlinear sampled-data control systems is scarce, but three different methods are described in this thesis; emulation design, direct discrete-time design, and sampled-data design.

The electro-pneumatic clutch actuator is controlled using a continuous-time backstepping controller implemented digitally. This is essentially the procedure of emulation design and is the common, if not only, method used in practical engineering tasks so far. However, redesign of the continuous-time controller using the direct discrete-time method shows great potential of improving performance and robustness of sampled-data systems. Direct discrete-time design is based on an approximate discrete-time model of the plant, giving the controller a structure that accounts for the sampling of the hybrid system. Potentially, one can utilize slower sampling in the system by implementing a discrete-time controller into the digial computer instead of a continuous-time one. Examples and case studies that prove the improvement one can achieve by chosing the direct discrete-time design is included in the first part of the thesis.

Both a third- and fifth-order model of the electro-pneumatic clutch actuator are presented, and used as a basis for continuous- and discrete-time state-feedback backstepping controllers. These controllers are simulated with different sampling intervals to show their performance under different circumstances. The continuous-time controllers prove good reference trajectory tracking of the pure continuous-time system, while the performance of the sampled-data systems descends as higher sampling intervals are used. And, as opposed to the mentioned examples and case studies, the controller designed when taking the sampling into account shows no sign to outperform the controller that was designed without considering the sampling, at least not for the relative fast sampling the clutch actuator operates with.

Contents

Chapter 1

Introduction

Kongsberg Automotive develops, manufactures and markets systems for gearshift, clutch actuation, seat comfort, stabilising rods, couplings and components. KA's clutch actuation systems provide important ergonomic and environment-friendly features that meet the automotive industry's performance requirements. In heavy duty trucks, automated manual transmission (AMT) systems consist of a electro-pneumatic actuator used to automate the clutch and gear shift operation.

The AMT systems use friction disc clutches consisting of two discs, where one is connected to the engine, and the other to the driveline. The amount of torque transferred between the engine and the driveline is controlled by the the distance between the discs, which makes accurate control of the position of the clutch essential for smooth operation of the vehicle. Large uncontrolled variations in transferred torque mostly appear during start and stop of the vehichle; if the clutch is engaged too fast oscillations may occur and the friction discs can be damaged. The lifetime of the clutch can be considerably improved by reducing this wear and tear of the friction discs, and good control of the clutch also ensures increased comfort for the driver.

Because pressurized air is available on the truck, pneumatic actuators are the preferable choice over hydraulic actuators due to the reduction in assembly cost. In addition, the pneumatic clutch actuators are easier to maintain and are more environmental friendly due to the risk of leakages in the standard hydraulic solutions. Although, the pneumatic actuators are inherently difficult to control compared to their hydraulic counterpart, because of the high compressibility and nonlinear flow characteristics of air.

1.1 System Description

The electro-pneumatic clutch actuator is mainly consisting of five parts; a clutch, an actuator, a valve, an electronic control unit (ECU) and a position sensor, as shown in figure 1.1. Information about the various parts of the system described below is obtained from Kaasa (2003).

Clutch: The clutch controls the engagement and disengagement of the engine from the transmission and the driveline, which is necessary during transmission gear shift and during start and stop of the vehicle. The fricion discs are compressed by a diaphragm spring and an actuator spring, ensuring the driveline is being connected to the engine through the transmission and the clutch. The amount of torque transferred through the clutch is controlled by the position of the friction plates relative to each other and the friction force generated by them.

Actuator: The actuator is a cylinder with a piston, where a lip seal is used to keep the air inside the two chambers apart. A higher pressure in chamber A than in chamber B renders a force which pushes the actuator towards positive position y. Hence, the position of the friction discs and the torque transferred through the clutch is controlled by the pressure p_a.

Valve: The pressure in chamber A of the actuator is controlled by a proportional valve. An air supply resevoir and an exhaust chamber are connected to the valve: the supply pressure is P_s, and the exhaust chamber is connected to atmospheric pressure P_0. The input voltage u_v controls the air mass flow rate into chamber A by controlling the internal valve piston which again controls the size of the orifice connecting either the supply or the exhaust chamber to the actuator.

Electronic Control Unit: The ECU is the digital computer controlling the clutch. This unit makes the electro-pneumatic actuator a hybrid system (see Chapter 2), as opposed to using an analog control unit. Measurements from the AMT enter the ECU, and it's output is controlling the valve input u_v.

Position sensor: The signal of the position sensor, which is inside the clutch, is sent back to the ECU and used as feedback control for the system.

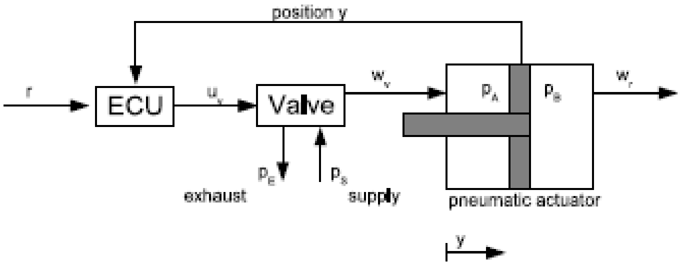


Figure 1.1: Schemating drawing of the electro-pneumatic actuator

1.2 Problem Statement

The control theory of nonlinear sampled-data systems is quite an underdeveloped subject. Controllers based on continuous-time design used in digital computers may not perform as well as in a pure continuous-time environment. The electro-pneumatic clutch actuator is a nonlinear sampled-data system, and it was KA's aim to investigate different possibilities of controller design, especially implementing their controller in a discrete-time domain to possibly better the performance and robustness of their system. The introduction of a discrete-time controller to their system might give the opportunity to use slower sampling, and still maintain approved performance.

This master thesis is based on a literature study worked out in the autumn 2006. During this literature study one method of control of nonlinear sampled-data systems was singled out as the most promising, and thus is the basis of this thesis.

The work do be done during the spring was to, as stated by the supervisors; design, test and compare controllers based on continuous and discrete backstepping. The design, test and comparison was at first to be conducted on some general examples, and then extracted to both a 3rd- and 5th-order model of the electro-pneumatic clutch actuator. The comparison of the controllers should demonstrate whether designing a controller in the discrete-time domain would be improving the performance of the system, and was to be conducted for many different sampling intervals.

1.3 Control Theory

The reader is assumed to be familiar with common nonlinear control theory and notation. An introduction to nonlinear systems as well as tools for stability analysis and control of these, is given in Khalil (1996).

Continuous-time backstepping is a well established method of control design that are based on elementary Lyapunov analysis. The main idea is to first control the output of the first subsystem by the use of a virtual control, and then step back through each integrator until the actual control input appears in the equation. The actual control input is then chosen to guarantee stability of the closed loop control system. The main assumption for the backstepping used in this thesis is that the system equations are on state feedback form. More on state feedback, and a detailed description of numerous methods of backstepping, can be viewed in Krstić et al. (1995).

The reader need not be familiar with nonlinear sampled-data systems and the control of these, since this is the subject of Chapter 2 and 3 of this thesis.

1.4 Outline of the Thesis

The thesis is arranged as follows:

Chapter 2 consists of a description of nonlinear sampled-data systems in general, together with an overview of the different methods of controller design for these systems, and examples of their use. The chapter is concluded with a simple case study to show the differences in performance of the various controllers, for different sampling intervals and initial conditions.

Chapter 3 gives a literature review and a detailed description of the direct discrete-time method used for controller design in this thesis.

Chapter 4 presents the third order mathematical model of the clutch actuator, as well as design of both continuous- and discrete-time controllers, and a comparison between these.

Chapter 5 presents the fifth order mathematical model of the clutch actuator, as well as design of both continuous- and discrete-time controllers, and a comparison between these.

Chapter 6 concludes the report, and presents ideas for further work.

Chapter 2

Nonlinear Sampled-Data Control Systems

Most control engineering systems developed nowadays use a digital computer to implement their chosen controller. Digitally implemented controllers outperform their analog counterpart in terms of cost, user friendliness, flexibility, expandability and simplicity. Since most plants found in engineering practice are continuous-time, this leads to a lot of systems with both continuous-time and discrete-time signals in their operation. These systems are called sampled-data systems to emphasize the sampling process as their crucial feature. A sampled-data control system is given schematically in figure 2.1.

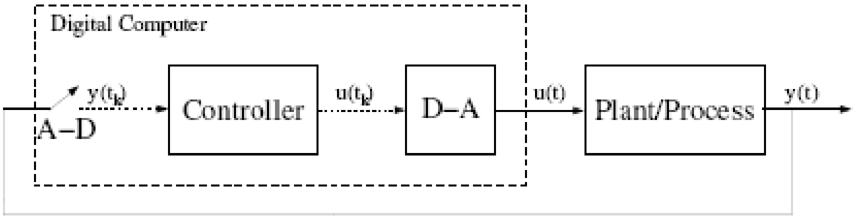


Figure 2.1: Sampled-data control systems

A continuous-time plant is interfaced with the computer via analog-to-digital (A/D) and digital-to-analog (D/A) converters; often referred to as *sampler* and *hold* devices, respectively. The A/D converter produces samples of the continuous plant output at sampling times, and sends them to the controller algorithm within the computer. The controller then processes the measured sequence, and produces a serie of control inputs which is converted in the D/A converter. The D/A converter produces a piecewise continuous control signal which is applied to the plant; this is usually done by holding the value of the control signal constant during the sampling intervals *(zero-order hold)*. An internal clock synchronizes the operation of the system.

Most plants and processes are nonlinear in nature. While it is possible to use a linear approximation around a prescribed operating point for analysis and controller design, there are many situations when nonlinearities can not, or should not, be neglected. A sampled-data control system which includes a nonlinear plant controlled by either a linear or nonlinear controller is classified as a nonlinear sampled-data control system. Due to the hybrid nature of nonlinear sampled-data systems, they are in general harder to analyze and design than pure continuous-time or discrete-time systems.

The process of transforming signals from analog to digital and vice versa is a crucial feature for this class of systems. Sampling is an essentiel operation involved in the transformation from analog to digital, and signal reconstruction is important for the converse. Clearly, some information carried by a signal is lost due to sampling, regardless of how good the reconstruction process is. Also, certain useful properties of continuous-time control systems, such as controllability and observability may be destroyed during sampling. Therefore, the developement of special tools to carry out analysis and design for nonlinear sampled-data systems is an important research area with a number of open problems.

2.1 Controller Design

According to existing literature, e.g. Laila (2003), there are three different ways to make controllers for nonlinear sampled-data systems. The first technique is called emulation design, and is the one currently used by the industry in general, and the electro-pneumatic clutch actuator specifically. Here the controller design is done in the continuous-time domain followed by a controller discretization to produce a discrete-time controller for digital implementation.

Direct discrete-time design is the second technique, where a discrete-time controller is designed in discrete-time domain directly, using an approximate discrete-time model of the plant. This technique shows potential to improve performance of nonlinear sampled-data control systems, and is the approach pursued in this thesis.

The third technique is called sampled-data design, which is a technique not fully developed yet. The following subsections contain a more detailed overview of the different techniques, as described in Laila (2003).

2.1.1 Emulation Design

While there exist a large variety of tools for continuous-time design, tools for discrete-time controller design for sampled-data systems are scarce. The use of the emulation technique was initiated by treating discrete-time systems in a continuous-time framework. Emulation is regarded as the simplest method of controller design for sampled-data systems, and is also most of the time inferior to the other two methods in terms of stability and performance. Emulation makes use of control design tools of continuous-time systems, and follows the steps shown in figure 2.2.

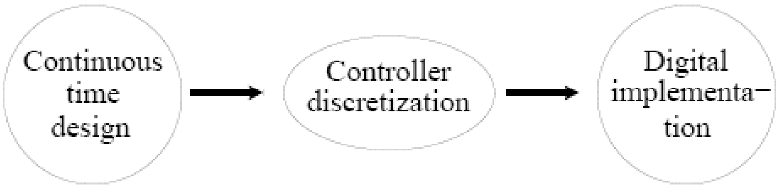


Figure 2.2: Emulation-design

The first step is to design a continuous-time controller, using any continuous-time design tool available in the literature. The obtained continuous-time controller should achieve a set of perfomance and robustness criteria for the closed-loop continuous-time system. At this step, sampling is completely ignored.

Secondly, the controller is discretized using some kind of discretization approximation. Fast sampling is required using this technique, since the approximate discrete-time model is a good approximation of the continuous-time model only for small sampling periods.

The last step of the emulation design is to implement the controller digitally, using sampler and hold devices as described above. The configuration of a sampled-data control system which is obtained from emulation design is given in figure 2.3. Here, K_T is a discretization of an originally designed continuous-time controller K_C, and S and H represent sampling and hold devices, respectively.

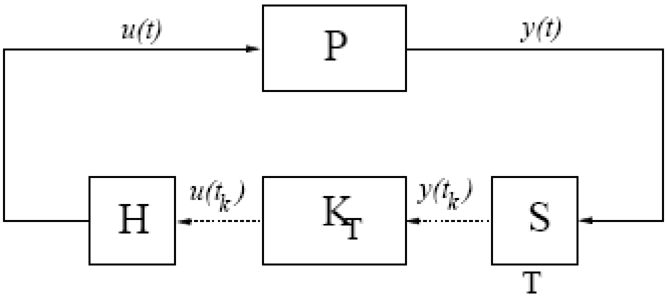


Figure 2.3: Control system configuration using emulation-design

The selection of the sampling period T is of great importance when using emulation design. It is necessary to choose a T low enough to preserve, in some sense, the stability and performance criteria for the continuous-time system. In the following example, parts of it obtained from Khalil (1996), emulation is applied to design a discrete-time controller for a continuous-time plant.

Example 1. *Consider the plant:*

$$\begin{aligned}\dot{x}_1 &= x_1^2 - x_1^3 + x_2 \\ \dot{x}_2 &= u.\end{aligned} \tag{2.1}$$

We use backstepping as the control tool, and start with the scalar system

$$\dot{x}_1 = x_1^2 - x_1^3 + x_2$$

with x_2 viewed as the input, and proceed to design a feedback control law to stabilize the origin $x_1 = 0$. With

$$x_2 = \phi(x_1) = -x_1^2 - x_1$$

we cancel the nonlinear term x_1^2 to obtain (we don't cancel $-x_1^3$ since it provides nonlinear damping)

$$\dot{x}_1 = -x_1 - x_1^3,$$

and $V(x_1) = \frac{1}{2}x_1^2$ satisfies

$$\dot{V} = -x_1^2 - x_1^4 < 0, \qquad \forall x_1 \neq 0.$$

Hence, the origin of $\dot{x}_1 = -x_1 - x_1^3$ *is globally exponentially stable. To backstep, we use the change of variables*

$$z_2 = x_2 - \phi(x_1) = x_2 + x_1 + x_1^2$$

to transform the system into the form

$$\begin{aligned}\dot{x}_1 &= -x_1 - x_1^3 + z_2\\ \dot{z}_2 &= u + (1 + 2x_1)(-x_1 - x_1^3 + z_2).\end{aligned}$$

Taking $V(x) = \frac{1}{2}(x_1^2 + z_2^2)$ *as the composite Lyapunov function, we obtain*

$$\begin{aligned}\dot{V} &= x_1(-x_1 - x_1^3 + z_2) + z_2[u + (1 + 2x_1)(-x_1 - x_1^3 + z_2)]\\ &= x_1^2 - x_1^4 + z_2[x_1 + (1 + 2x_1)(-x_1 - x_1^3 + z_2) + u]\\ &= -x_1^2 - x_1^4 - z_2^2 < 0 \ \ \forall x_1, z_2 \neq 0\end{aligned}$$

when using the following control law

$$u_{ct} = -(2x_1 + 1)(x_1^2 - x_1^3 + x_2) - x_1 - (x_2 + x_1^2 + x_1). \tag{2.2}$$

With the controller u_{ct} *we achieve global asymptotic stability of the origin of the closed loop continuous-time system* (2.1), (2.2). *The discretization is done using a sample and zero-order hold; by holding the control value constant between every sampling period;*

$$u(k) = -(2x_1(k) + 1)(x_1(k)^2 - x_1(k)^3 + x_2(k)) - x_1(k) - (x_2(k) + x_1(k)^2 + x_1(k)) \tag{2.3}$$

For a sufficient small sampling period T, the closed loop sampled-data system (2.1), (2.3) *is asymptotic stable, with ultimate boundedness in the states, which is known as the semiglobal practical property according to Laila (2003).* ■

Emulation controllers work well under fast sampling, thus choosing a smaller sampling period can improve performance of emulation controlled systems. Improvement can also be achieved by using a better discretization technique, or by redesigning the controller by using direct discrete-time design to obtain some improvements. This last technique may give a controller which takes the form

$$u_{dt} = u_{ct} + Tu_1,$$

where u_{ct} is the continuous-time controller and u_1 is the additional term obtained from redesigning u_{ct}. The redesign, which is described in Nesic et al. (2005), is typically done by using the continuous-time Lyapunov function V as a control Lyapunov function for an approximate discrete-time model F_T^a that is one step consistent (see Appendix A for definiton of one step consistency) with the exact model F_T^e. That is, we consider:

$$\frac{V(F_T^a(x, u_{ct} + Tu_1)) - V(x)}{T}$$

where F_T^a is one step consistent with F_T^e, and u_{ct} and V were obtained from an arbitrary continuous-time design, as in the example above. One way to design u_1 is to require that:

$$\frac{V(F_T^a(x, u_{ct} + Tu_1)) - V(x)}{T} < \frac{V(F_T^a(x, u_{ct})) - V(x)}{T}. \tag{2.4}$$

In other words, u_1 can be designed to achieve more decrease for the Lyapunov function along solutions of the closed-loop approximate model with the redesigned controller. If u_1 is designed to satisfy (2.4) and it is bounded on compact sets, it can be concluded that the sampled-data system with the redesigned controller is SPA stable (see Appendix A). This approach may outperform the emulated controller, especially for larger sampling periods, as will be shown in the next section.

There are a number of advantages of the emulation design technique compared to the more complex methods. The tools for controller design in continuous-time are well established, and the design separates the controller design problem from the issue of choosing a sampling period. Since most plants are continuous, this design is a natural and favored method, and is used quite extensively in engineering pratice. The disadvantage that limits the usefulness of this method however, is the need for very fast sampling.

2.1.2 Direct Discrete-Time Design

Very fast sampling may in many cases not be feasible because of hardware limitations. Direct discrete-time design offers an alternative solution, where sampling is considered from the beginning of the design process. This type of design often shows potential to obtain controllers that improve performance of the closed-loop sampled-data systems, especially for higher sampling periods, which will be shown later in this chapter.

Direct discrete-time design is done directly in discrete-time domain, based on the discrete-time model of the plant. Regarding the plant model, there are two approaches taken in the available literature; assuming that the exact discrete-time model of the plant is known, or assuming that the exact model is unknown. Whereas the assumption that the exact discrete-time model is known usually holds for linear systems, it almost never holds for nonlinear systems. In general the exact discrete-time model of a nonlinear plant can not be computed from the continuous-time model since it needs an explicit analytic solution of a nonlinear differential equation, which is in general impossible to obtain. Because of this, approximate models are most commonly used in practice.

To obtain the approximate discrete-time model from the continuous-time model, numerical methods are used, even for linear systems. This is because numerical methods always preserves the feedback structure that may be needed for the controller design, and a simple Euler model is the most commonly used approximation technique. Since design is carried out based on the approximate plant model, there is no guarantee for the stability of the exact model, and therefore one needs to check the validity of the design based on several conditions.

The procedure of approximate direct discrete-time design is carried out in three steps, as sketched in figure 2.4.

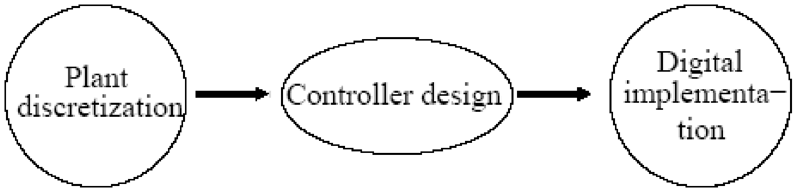


Figure 2.4: Direct discrete-time design

The first step of the design is the discretization of the continuous-time model in order to obtain the approximate discrete-time model of the plant. The discrete-time model is usually parameterized by the sampling period T, which often is left as a parameter to be determined later. The second step is to find a discrete-time controller for the plant, so that the closed-loop discrete-time system satisfies the stability and robustness criteria that have been set for the design. At this stage T is determined so that a satisfactory performance for the system is achieved.

However, when using direct discrete-time design, the intersample behaviour of the system is ignored. As a result, the stability of the approximate discrete-time model does not automatically imply stability of the original sampled-data system. The stability property of the closed-loop sampled-data system is determined based on certain conditions on the discrete-time model, the controller and the property achieved for the discrete-time model. Therefore, design verifications need to be done before implementing the controller to the original continuous-time plant. In Chapter 3 the direct discrete-time design method is investigated closer.

In the following example, partly obtained from Laila (2003), approximate discrete-time design is applied to obtain a discrete-time controller for a continuous-time plant, when an Euler-based discrete-time backstepping method is used as the design tool.

Example 2. *The Euler approximate model of the plant* (2.1)*:*

$$\begin{aligned} x_1(k+1) &= x_1(k) + T(x_1(k)^2 - x_1(k)^3 + x_2(k)) \\ x_2(k+1) &= x_2(k) + Tu(k) \end{aligned} \tag{2.5}$$

Applying discrete-time backstepping as described in Chapter 3, we have the following design parameters:

$$\begin{aligned} \alpha_T &= -x_1 - x_1^2 \\ W_T &= \frac{1}{2}x_1^2. \end{aligned}$$

We then compute (for simplicity, we omitt the argument (k)

$$\begin{aligned} \Delta\alpha_T &= -(x_1 + T(x_1^2 - x_1^3 + x_2)) - (x_1 + T(x_1^2 - x_1^3 + x_2))^2 + (x_1 + x_1^2) \\ &= -x_1 - T(x_1^2 - x_1^3 + x_2) - x_1^2 - 2Tx_1(x_1^2 - x_1^3 + x_2) - T^2(x_1^2 - x_1^3 + x_2)^2 + x_1 + x_1^2 \\ &= -T(2x_1 + 1)(x_1^2 - x_1^3 + x_2) - T^2(x_1^2 - x_1^3 + x_2)^2 \end{aligned}$$

and

$$\begin{aligned} \overline{\Delta W}_T &= \frac{1}{2}(x_1 + T(x_1^2 - x_1^3 + x_2))^2 - \frac{1}{2}(x_1 + T(x_1^2 - x_1^3 + \alpha_T))^2 \\ &= \frac{1}{2}[2x_1 + T(2x_1^2 - 2x_1^3 + x_2 + \alpha_T)] \cdot [T(x_2 - \alpha_T)] \end{aligned}$$

which gives

$$\widetilde{\Delta W}_T = \frac{\overline{\Delta W}_T}{x_2 - \alpha_T} = T\left[x_1 + T\frac{1}{2}(-x_1 + x_1^2 - 2x_1^3 + x_2)\right].$$

Then, the final control law is:

$$\begin{aligned}
u_{dt} &= -(x_2 + x_1 + x_1^2) - \frac{\widetilde{\Delta W}_T}{T} + \frac{\Delta\alpha_T}{T} \\
&= -(x_2 + x_1 + x_1^2) - x_1 - T\frac{1}{2}(-x_1 + x_1^2 - 2x_1^3 + x_2) - (2x_1 + 1)(x_1^2 - x_1^3 + x_2) - T(x_1^2 - x_1^3 + x_2)^2 \\
&= u_{ct} - T\left[(x_1^2 - x_1^3 + x_2)^2 + \frac{1}{2}(-x_1 + x_1^2 - 2x_1^3 + x_2)\right].
\end{aligned} \tag{2.6}$$

As shown in the above equation, the discrete-time control law u_{dt} *is in reality the continuous-time control law* (2.2) *in addition to a term multiplied with the sampling interval* T. *This is always the case when using this method to obtain a discrete-time backstepping controller. The composite Lyapunov function*

$$V_T(x) = \frac{1}{2}x_1^2 + \frac{1}{2}(x_2 + x_1^2 + x_1)^2$$

proves that the controller achieves semiglobal practical stability (see Appendix A) for the closed-loop approximate discrete-time model (2.5), (2.6), *and it can be shown further that it also asymptotically stabilizes the closed-loop sampled-data system* (2.1), (2.6) *in a semiglobal practical sense (see Chapter 4).* ■

Obviously, analysis and design using this method suffer from the fact that the discretization of the plant may destroy some important properties of the continuous-time model, such as the loss of feedback linearizability and minimum phase properties. Consequently, analysis and design using the direct discrete-time method is usually harder, which leads to the fact that emulation design still are pretty much the only used technique in practice. Whilst there are some results that prove the potential of direct discrete-time design, this has, as far as the writer knows, yet to be implemented in practice on any physical systems.

2.1.3 Sampled-Data Design

It was mentioned above that direct discrete-time design potentially improves the controller performance compared to the emulation method, but since intersample behaviour is not taken into account in the direct discrete-time design, ripple may occur in the response of the system. To guarantee good performance of the controlled systems, careful design and choice of sampling period have to be done. One way to take the intersample behaviour into account is the sampled-data design, which is the third approach of controller design for sampled-data systems.

This method involves using the exact sampled-data model of the plant, and designing a controller that achieves both stability and required performance for the sampled-data system. This method uses no approximation of the plant or the controller, hence it maintains stability and performance for arbitrarily large sampling periods T.

The theory of sampled-data design for linear systems is quite developed, e.g. Chen et al. (1995). Because of the complexity of the underlying nonlinear sampled-data model, results on this design for nonlinear systems are scarce, and it appears that they will be hard to develope in the future.

It appears that discrete-time design techniques for nonlinear-sampled data systems provide a nice tradeoff between the possible conservatism of emulation design and the difficulty of developing direct sampled data-design methods. Hence, the direct discrete-time design is the approach that will be pursued in this thesis.

2.2 A Simple Case Study

This section presents a simple case study where a continuous-time plant is controlled both by a continuous-time controller, and three different discrete-time controllers. First the controller designs are presented, then the controllers are compared for different sampling intervals and different initial conditions to prove that the emulation design is sometimes inferior to designs which account for the sampling of the system.

Consider the nonlinear continuous-time plant

$$\begin{aligned}\dot{\eta} &= \eta^2 + \xi \\ \dot{\xi} &= u.\end{aligned} \tag{2.7}$$

We start the controller designs with presenting the continuous-time controller, which is the basis for the three discrete-time controllers designed thereafter.

2.2.1 Continuous-Time Design

Consider the continuous-time system (2.7). To design a continuous-time backstepping controller, the same procedure as in Subsection 2.1.1 is followed. The first subsystem can be stabilized when using the control

$$\xi = \phi(\eta) = -\eta^2 - \eta$$

with the Lyapunov function $V(\eta) = \frac{1}{2}\eta^2$. Using this information and applying Krstić et al. (1995, Lemma 2.8 with c = 1), we obtain

$$u_{ct} = -2\eta - \eta^2 - \xi - (2\eta + 1)(\eta^2 + \xi) \tag{2.8}$$

which globally exponentially stabilizes the origin of the closed loop continuous-time system (2.7), (2.8) with the Lyapunov function $V(\eta, \xi) = \frac{1}{2}\eta^2 + \frac{1}{2}(\xi + \eta + \eta^2)$. Next, the simple emulation-controller is presented.

2.2.2 Emulation Design

The emulation controller is obtained by a plain discretization of the continuous-time controller (2.8):

$$u_{em} = u_{ct}(k) = -2\eta(k) - \eta^2(k) - \xi(k) - (2\eta(k) + 1)(\eta^2(k) + \xi(k)) \tag{2.9}$$

which achieves semiglobal-practical asymptotic (SPA) stability for the closed loop sampled-data system (2.7), (2.9) according to Theorem 5 (Appendix A) for sufficient small sampling intervals. The controller is implemented using sampler and zero-order hold devices as described earlier. From this point, for the sake of simplicity, the argument (k) is omitted whenever we are talking about discrete-time controllers. Following, the direct discrete-time design is used to obtain a discrete-time controller based on the Euler-approximate model of the system.

2.2.3 Direct Discrete-Time Design

The Euler approximate model of (2.7) is

$$\begin{aligned} \eta(k+1) &= \eta(k) + T(\eta^2(k) + \xi(k)) \\ \xi(k+1) &= \xi(k) + Tu(k). \end{aligned} \tag{2.10}$$

The parameters for the design, which are obtained from the first step of the continuous-time backstepping procedure, are

$$\begin{aligned} \alpha_T(\eta) &= -\eta^2 - \eta \\ W_T(\eta) &= \frac{1}{2}\eta^2. \end{aligned}$$

Following the technique presented in Section 3.2, we obtain

$$\Delta\alpha_T = -T(2\eta+1)(\eta^2+\xi) - T^2(\eta^2+\xi)^2$$

and

$$\widetilde{\Delta W}_T = T\left(\eta + \frac{T}{2}(\eta^2 - \eta + \xi)\right)$$

which, with c=1 give the control law

$$u_{dt} = u_{ct} - T\left[\frac{1}{2}(\eta^2 - \eta + \xi) + (\eta^2+\xi)^2\right], \tag{2.11}$$

which by Theorem 3 SPA stabilizes the closed-loop approximate discrete-time system (2.10), (2.11), which can be proven with the Lyapunov function $V(\eta,\xi) = \frac{1}{2}\eta^2 + \frac{1}{2}(\xi+\eta+\eta^2)^2$. By Theorem 5, the same controller u_{dt} SPA stabilizes the exact discrete-time model and consequently the closed loop sampled-data system (2.7), (2.11). The fourth controller is designed using the Lyapunov redesign method, which is briefly described in Subsection 2.1.1.

2.2.4 Lyapunov Redesign

Consider again the Euler approximate model (2.10). Denote $x := (\eta \;\; \xi)^T$. Suppose that $u_{re}(x) = u_{ct}(x) + Tu_1(x)$, then, according to Laila et al. (2005)

$$\frac{V(F_T^{Euler}(x, u_{ct}(x) + Tu_1)) - V(x)}{T} = \eta^2 - (\xi+\eta+\eta^2)^2 + Tp_1(u_1,x) + O(T^2)$$

where

$$p_1(u_1,x) = \frac{1}{2}(\eta^2+\xi^2)^2 + (\xi+\eta+\eta^2)(u_1 + (\eta^2+\xi)^2) + \frac{1}{2}(2\eta+\eta^2+\xi)^2,$$

and $O(T^2)$ contains higher order terms in T. Since T will have to be chosen small, $O(T^2)$ is neglected, and u_1 is chosen so that the term $p_1(u_1,x)$ is made more negative (note that there are some terms that can not be made more negative using u_1). One obvious choice is

$$u_1(x) = -(\eta^2+\xi)^2 - (\xi+\eta+\eta^2),$$

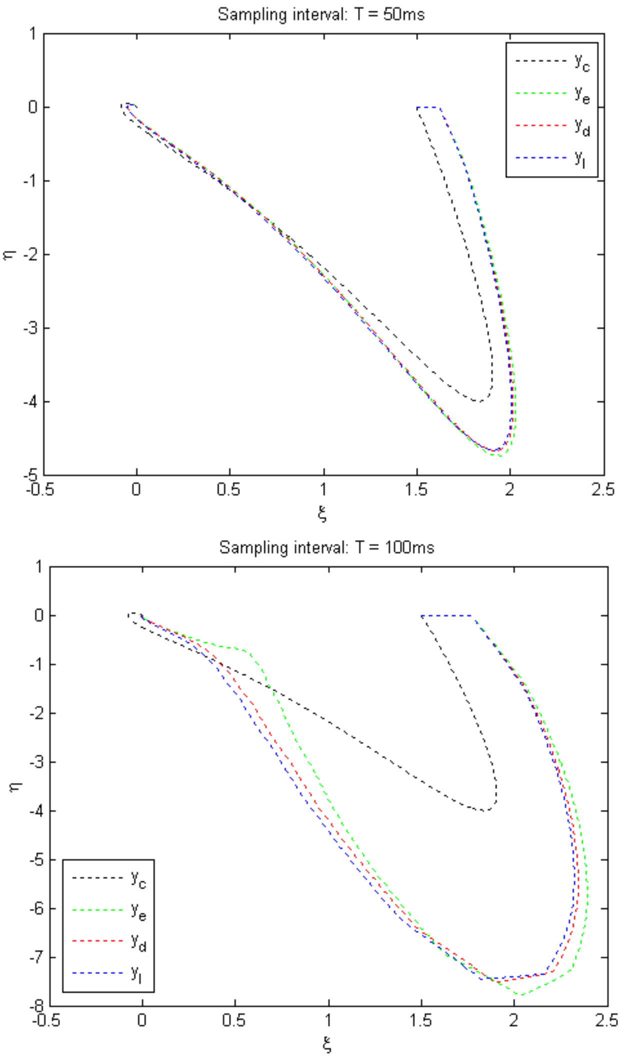
which cancels one term and provides extra damping. The redesigned controller is then

$$u_{re} = u_{ct} - T\left[(\eta^2 + \eta + \xi) + (\xi + \eta^2)^2\right], \tag{2.12}$$

which is designed to achieve more decrease for the Lyapunov function along solutions of the closed-loop approximate model with the redesigned controller. u_{re} is satisfying (2.4) and bounded on compact sets, thus from Theorem 5 the closed-loop sampled data-system (2.7), (2.12) is SPA stable.

2.2.5 Controller Comparison

The four controllers were simulated with different sampling intervals, and the response of the system is given in the following plots. In the first example the initial value is set to $x_0 = [1.5\ 0]$, and as shown in figure 2.5 all the trajectories are bounded and converges the origin when the sampling interval is $T = 50ms$, $T = 100ms$ and $T = 150ms$. In all the plots y_c means the system with the continuous-time controller, y_e the emulation controller, and y_d and y_l the systems with controllers obtained by direct discrete-time design and Lyapunov redesign, respectively.



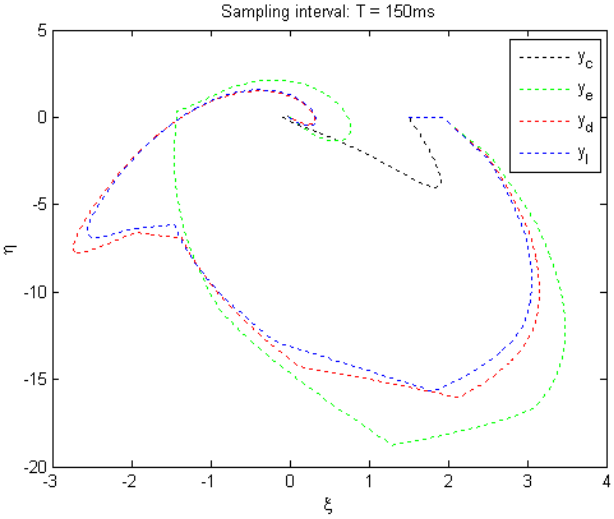


Figure 2.5: Simulations at 50ms, 100ms and 150ms sampling interval

Raising the sampling interval to $T = 200ms$ as in figure 2.6, with the same initial condition, the trajectory of the emulation-controlled system escapes in finite time, while the others still are bounded and converge to the origin. The response of the pure continuous-time system is obvioulsy the same, as it is not affected by the different sampling intervals. From this plot it can even seem that the Lyapunov redesigned controller in some sense outperforms the direct discrete-time controller for higher sampling periods. Although, due to the complexity of the Lyapunov redesign, this approach will not be pursued after this case study.

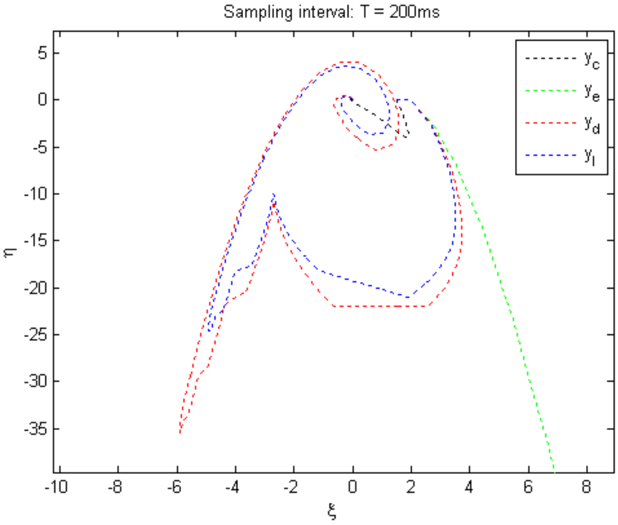


Figure 2.6: Simulations at 200ms sampling interval

Another example is presented in figure 2.7, where the initial value is set to $x_0 = [2\ 2]$. At a sample interval of $T = 90ms$, the trajectories of the two systems with controllers designed in the discrete-time domain converges to origo in six seconds time, while the emulation controller clearly destabilizes the system.

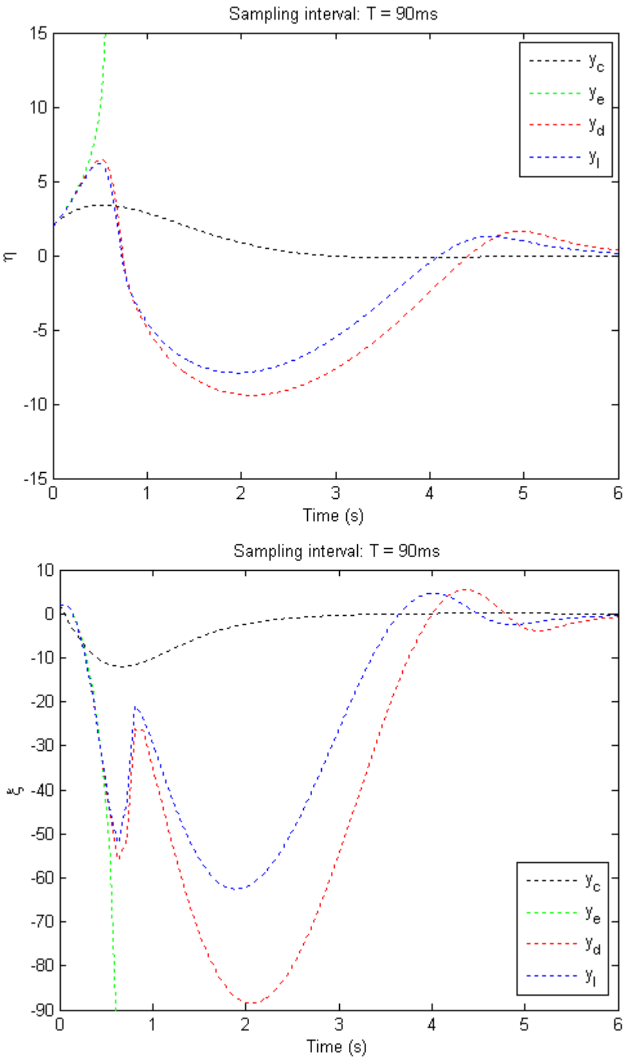


Figure 2.7: Simulations at 90ms sampling interval

The last example is taken from Nesic et al. (2006) and shows the trajectories of the emulation-controlled system and the system with the Euler-based controller, for a sampling interval of $T = 500ms$ and an initial value of $x_0 = [1.6\ 0]$ (figure 2.8). Also, domain of attraction (DOA) estimates of the two controllers for the same sampling interval are shown in figure 2.9. Here, stars represent the region of attraction of the emulation-controller, and triangles represent the region of attraction of the Euler-based controller.

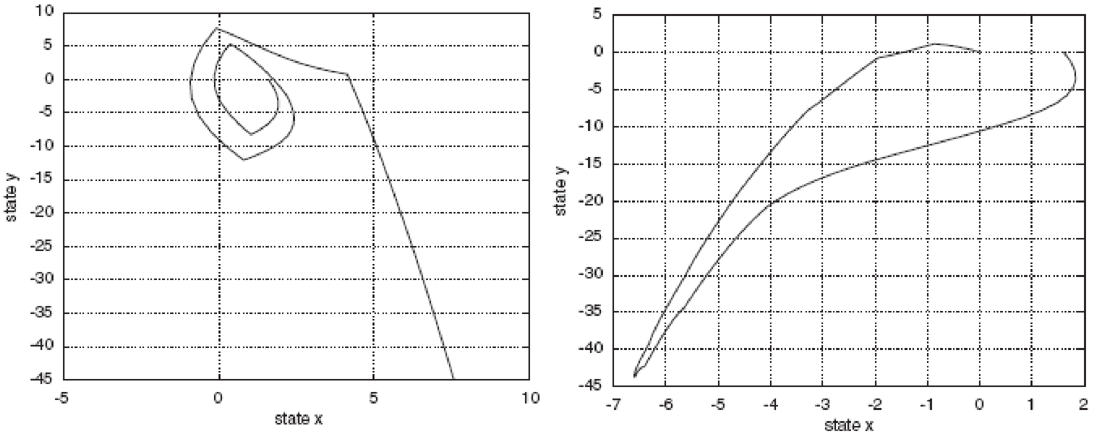


Figure 2.8: Emulation- and euler-based controller, respectively, for 500ms sampling interval

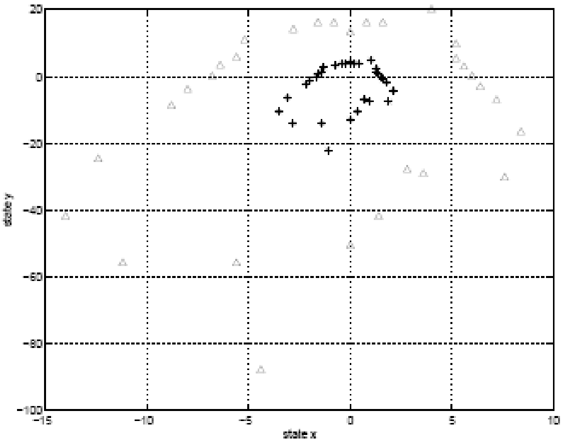


Figure 2.9: Domain of attraction estimates for $T = 500ms$

Nesic et al. (2006) claims that u_{dt} constantly yields four times larger DOA than u_{ct} for all tested sampling periods on this system. From figure 2.8 we see again that with the given sampling time and initial value, one controller achieves stability and boundedness while the other does not.

By these examples it is quite clear that by taking the sampling of the nonlinear sampled-data system into account when designing the controller, there are plenty to gain in terms of stability, especially for higher sampling intervals. Therefore, it is proved that the direct discrete-time design (and the Lyapunov redesign) has the possibility of improving the behaviour of nonlinear sampled-data control systems, and thus merits the choice of pursuing this approach. The next chapter presents the direct discrete-time method in detail, and in the rest of this thesis the available theory is extracted and used to design controllers for the electro-pneumatic clutch actuator.

Chapter 3

Direct Discrete-Time Design

In this chapter the technique for discrete-time controller design based on the Euler-approximate model of the system is investigated closer. We start with a brief literature review of the available theory on the direct discrete-time method.

3.1 Literature Review

Nesic et al. (2000) proves that, if some general conditions holds, the family of controllers that stabilize the approximate discrete-time model of a plant also guarantees stabilization of the exact discrete-time model in a semiglobal-practical sense for sufficient small sampling periods. Furthermore, if the family of controllers is locally bounded, uniformly in the sampling period, the intersample behavior can also be uniformly bounded, so that the (time-varying) sampled-data model of the plant is uniformly semiglobally-practically stabilized. This result justifies the controller design for nonlinear sampled-data system based on the approximate discrete-time model of the plant.

The direct discrete-time method itself is first presented in Nesic et al. (2001), where two integrator backstepping designs are presented for digitally controlled continuous-time plants in feedback form. The controller designs are based on the Euler approximate model of the plant, and the two control laws yield, respectively, semiglobal-practical stabilization and global asymptotic stabilization of the Euler model. Both designs achieve semiglobal-practical stabilization (in the sampling period that is regarded as a design parameter) of the closed-loop sampled-data system. The motivation for the backstepping designs based on the Euler approximate model is:

1. The Euler approximate discrete-time model preserves the strict feedback structure of the continuous-time plant.
2. The attainement of completely new control algorithms.
3. The backstepping controllers based on the Euler approximate model may outperform discretized continuous-time backstepping controllers implemented digitally.
4. A careful investigation of the design is needed to ensure that it is valid for given circumstances.

Nesic et al. (2004) gathers all the abovementioned results in a unified framwork for design of stabilizing controllers for nonlinear sampled-data systems via their approximate discrete-time models. Both fixed and fast sampling are considered.

Laila (2003) is an extensive PhD thesis that consists of a general and unified framework for both the emulation- and direct-discrete time design, together with case studies to illustrate an engineering application with the results on the controller designs. Input-to-state stabilization of a two-link manipulator is considered, and it is demonstrated that a controller designed using direct discrete-time design may outperform the controller designed using emulation.

Nesic et al. (2006) is a more complete version of the paper from 2001, with extended proofs of the integrator backstepping designs, and examples of their use.

3.2 Preliminaries

This section states preliminary results that are used in the controller design method in the next section.

Consider the continuous-time system

$$\begin{aligned} \dot{\eta} &= f(\eta) + g(\eta)\xi \\ \dot{\xi} &= u. \end{aligned} \tag{3.1}$$

Consider the difference equations corresponding to the exact plant model and its Euler approximation, respectively:

$$x(k+1) = F_T^e(x(k), u(k)) \tag{3.2a}$$

$$x(k+1) = F_T^{Euler}(x(k), u(k)), \tag{3.2b}$$

where the notation $x := (\eta^T \xi^T)^T$ is used, and

$$F_T^{Euler}(x,u) := \begin{pmatrix} \eta + T(f(\eta) + g(\eta)\xi) \\ \xi + Tu \end{pmatrix}.$$

Both models (3.2a) and (3.2b) are parameterized with the sampling period T. F_T^e is not known in most cases, and even if it is it usually does not have the feedback structure necessary to perform backstepping, which F_T^{Euler} always has. In general, one needs to use small sampling period T since the approximate plant model is a good approximation typical only for small T. Clearly, the goal is to obtain a controller $u_T(x)$ based on the Euler model which is, in general, parameterized by T and which is defined for small T. The following definitions are used, as stated in Nesic et al. (2006).

Definition 1. *We say that the family of controllers u_T semiglobally-practically asymptotically (SPA) stabilizes F_T if there exists $\beta \in \mathcal{KL}$ such that for any pair of strictly positive real numbers (D, ν) there exists $T^\star$ such that for each $T \in (0, T^\star)$ the solutions of $x(k+1) = F_T(x(k), u_T(u(k)))$ satisfy:*

$$|x(k, x(0))| \leq \beta(|x(0)|, kT) + \nu, \quad k \geq 0,$$

whenever $|x(0)| \leq D$. ■

Definition 2. *Let $\hat{T} \geq 0$ be given and for each $T \in (0, \hat{T})$ let functions $V_T : \mathcal{R}^n \to \mathcal{R}_{\geq 0}$ and $u_T : \mathcal{R}^n \to \mathcal{R}$ be defined. We say that the pair (u_T, V_T) is a SPA stabilizing pair for F_T if there exists $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$ such that for any pair of strictly positive real numbers (Δ, δ) there exists a triple of strictly positive real numbers $(T^\star, L, M)$, with $T^\star \leq \hat{T}$, such that for all $x, z \in \mathcal{R}^n$ with* *max* $\{|x|,|z|\} \leq \Delta$ *and* $T \in (0, T^\star)$ *we have:*

$$\begin{aligned} \alpha_1(|x|) \leq V_T(x) &\leq \alpha_2(|x|) \\ \Delta V_T &\leq -T\alpha_3(|x|) + T\delta \\ |V_T(x) - V_T(z)| &\leq L|x-z| \\ |u_T(x)| &\leq M, \end{aligned}$$

where $\Delta V_T := V_T(F_T(x, u_T(x))) - V_T(x)$. *Moreover, if there exists* $T^{\star\star} > 0$ *such that the above conditions with* $\delta = 0$, *hold for all* $x \in \mathcal{R}^n$ *and all* $T \in (0, T^{\star\star})$ *then we say that the pair* (u_T, V_T) *is a globally asymptotically (GA) stabilizing pair for* F_T. ■

A direct consequence of Definition 2 is

Lemma 1. (u_T, V_T) *is a GA stabilizing pair for* $F_T \Rightarrow (u_T, V_T)$ *is a SPA stabilizing pair for* F_T. ■

The following two results come directly from Nesic et al. (2000).

Theorem 1. (u_T, V_T) *is a SPA stabilizing pair for* $F_T^{Euler} \Rightarrow (u_T, V_T)$ *is a SPA stabilizing pair for* F_T^e. ■

Theorem 2. (u_T, V_T) *is a SPA stabilizing pair for* $F_T \Rightarrow u_T$ *SPA stabilizes* F_T. ■

We now have the framework for the controller design, and can move on to the design process itself.

3.3 SPA Stabilizing Pair for the Euler Approximate Model

The Euler approximate model of (3.1) has the following form

$$\eta(k+1) = \eta(k) + T\left[f(\eta(k)) + g(\eta(k))\xi(k)\right] \tag{3.3a}$$

$$\xi(k+1) = \xi(k) + Tu(k). \tag{3.3b}$$

Then, from Nesic et al. (2001), we present the theorem to design a SPA stabilizing pair (u_T, V_T) for the Euler approximate model.

Theorem 3. *Consider the Euler approximate model* (3.3a), (3.3b). *Suppose that there exists* $\hat{T} \geq 0$ *and a pair* (α_T, W_T) *that is defined for all* $T \in (0, \hat{T})$ *and is a SPA stabilizing pair for the subsystem* (3.3a), *with* $\xi \in \mathcal{R}$ *regarded as control. Moreover, suppose that the pair* (α_T, W_T) *has the following properties.*

1. α_T *and* W_T *are continuously differentiable for any* $T \in (0, \hat{T})$.

2. *there exists* $\tilde{\varphi} \in \mathcal{K}_\infty$ *such that*

$$|\alpha_T(\eta)| \leq \tilde{\varphi}(|\eta|).$$

3. *For any $\tilde{\Delta} > 0$ there exists a pair of strictly positive numbers $(\tilde{T}, \tilde{M})$ such that for all $T \in (0, \tilde{T})$ and $|\eta| \leq \tilde{\Delta}$ we have*

$$max\left\{\left|\frac{\partial W_T}{\partial \eta}\right|, \left|\frac{\partial \alpha_T}{\partial \eta}\right|\right\} \leq \tilde{M}$$

Then, there exists a SPA stabilizing pair (u_T, V_T) for the Euler model (3.3a)*,* (3.3b)*. In particular, we can take*

$$u_T = -c(\xi - \alpha_T(\eta)) - \frac{\widetilde{\Delta W}_T}{T} + \frac{\Delta\alpha_T}{T} \tag{3.4}$$

where $c > 0$ is arbitrary and

$$\begin{aligned}
\Delta\alpha_T &:= \alpha_T(\eta + T(f + g\xi)) - \alpha_T(\eta) \\
\widetilde{\Delta W}_T &:= \begin{cases} \frac{\overline{\Delta W}_T}{(\xi - \alpha_T(\eta))}, & \xi \neq \alpha_T(\eta) \\ T\frac{\partial W_T}{\partial \eta}(\eta + T(f + g\xi))g, & \xi = \alpha_T(\eta) \end{cases} \\
\overline{\Delta W}_T &:= W_T(\eta + T(f + g\xi)) - W_T(\eta + T(f + g\alpha_T))
\end{aligned}$$

and the Lyapunov function

$$V_T(\eta, \xi) = W_T(\eta) + \frac{1}{2}(\xi - \alpha_T(\eta))^2.$$

■

Complete proof of this teorem is provided in Nesic et al. (2006). This Euler-based discrete-time backstepping approach achieves semiglobal practical asymptotic stabilization of the sampled-data system. In Nesic et al. (2000), the results that justify the design of nonlinear sampled-data controllers based on approximate discrete-time models are presented. It is shown that with the appropriate combination of consistency of the approximate discrete-time model with the exact discrete-time model, and stability uniformity in the sampling period for the family of approximate discrete-time closed loops, semiglobal-practical asymptotic stability of the origin of the sampled-data control system is guaranteed (see Appendix A). Like in the continuous-time case, separate tools like small gain theorems and Lyapunov conditions should be used to analyze *local* asymptotic stability properties.

3.3.1 Nonautonomous Systems

The technique described above is designed mainly for nonlinear system *without* time varying reference inputs, and there are no known results or examples in the available literature on nonautonomous systems. Although, following the proof of the method in Nesic et al. (2006), there is nothing which suggests that it is not applicable also for nonlinear systems with time varying reference inputs. This is backed up by the simulation results that are shown in the next two chapters. With this in mind, we now leave the theory behind and move on to the third order model of the electro-pneumatic clutch actuator.

Chapter 4

3rd Order Model

This chapter presents a simplified third order model of the electro-pneumatic actuator, together with both continuous- and discrete-time controller designs and a comparison between these. This chapter is meant as an indicator on how well the different controllers behave for different sampling intervals, and is used as a basis to the chapter on the fifth order model that is presented later.

4.1 Mathematical Model

The electro-pneumatic actuator can be mathematically modelled as

$$\begin{aligned} \dot{y} &= v \\ M\dot{v} &= -f_l(y) - f_f(v) + A\left[p(y,\zeta) - P_0\right] \\ \dot{\zeta} &= RT_0\omega, \end{aligned} \tag{4.1}$$

where y and v are the position and velocity of the actuator piston, respectively. The pressure in chamber A is $p(y,\zeta) = \frac{\zeta}{V(y)}$, where the initial volume $V(y) = V_0 + Ay$. The variable ζ is proportional to the accumulated air in the actuator, and represents the work done on the piston.

The load characteristic $f_l(y)$ is given by a parameter affine approximation with gaussian basis functions as explained in Kaasa (2003):

$$f_l(y) = \boldsymbol{\phi}_l^T(y)\boldsymbol{\theta}_l$$

Here $\boldsymbol{\theta}_l$ is a vector of scalar weights, and $\boldsymbol{\phi}_l^T(y)$ is a vector containing the basis functions. Gaussian basis functions are defined by:

$$\phi_i(y) = e^{-\boldsymbol{\omega}_i^2(y_l - c_i)^2}$$

where $\boldsymbol{\omega}$ and $\boldsymbol{c}$ are vectors with scaling and offset parameters, respectively.

In this simplified third order model, the friction dynamic is modelled as a viscous friction $f_f(v) = D_v v$. Furthermore, M and A are the mass and areal of the actuator piston, R is the gas constant, and P_0 and T_0 are the reference pressure and temperature, respectively. The mass flow ω is treated as the control input of the system. All the parameters are obtained from Løkken (2006), and can be viewed in table 4.1.

Parameter	Value	Unit
M	10	kg
A	0.1123	m^2
P_0	10^5	Pa
R	288	J/(kgK)
T_0	293	K
V_0	0,0008	m^3
D_v	5000	Ns/m

Table 4.1: Parameters used for the third order model

Some simplifications are made to the system in order to obtain the third order model, like:

- Isothermal conditions are assumed for the actuator; the temperatures in chamber A and B, the supply and exhaust temperatur are all assumed constant and equal to the reference temperature T_0.
- The pressure in chamber B is assumed to be constant and equal to the atmospheric pressure P_0.
- The friction force f_f is as mentioned modelled as a simple linear viscous friction, whereas it in reality is highly unlinear, and better modelled with the Lugre friction model.
- To get the model to third order, we view the mass flow as the control input, instead of modelling the flow and viewing the input to the valve as our control input.

The complete sixth order model of the electro-pneumatic actuator can be viewed in Kaasa (2003), whilst some simplified fifth order models are presented in Løkken (2006), one of them used in the next chapter of this thesis.

4.1.1 Region of Validity

Due to the physical limitations in the clutch actuator, there exists a feasible region for the state variable $x = [y \;\; v \;\; \zeta]^T$ and the input ω. The physical limits of the system are given by:

- The position y is limited due to the construction of the actuator; $y_{ub} = 25mm$ and $y_{lb} = 0mm$.
- KA's given bounds on the velocity v are; $v_{ub} = 0.2m/s$ and $v_{lb} = -0.2m/s$
- $\zeta = V(y)p$ and its limits can be calculated since reasonable boundaries for the pressure p are given by the supply and exhaust pressure; $p_{ub} = P_S$ and $p_{lb} = P_0$
- The mass flow ω is bounded by the capacity of the valve, and can be calculated as $\omega_{ub} = \rho_0 P_S C_v \approx 0.03kg/s$

The feasible region for the system states is referred to as the *region of validity* and is given by

$$\Omega_v = \{\forall x, \forall \omega | x_{LB} \leq x \leq x_{UB} | \omega_{lb} \leq \omega \leq \omega_{ub}\}$$

where the upper and lower boundaries are defined as

$$\begin{aligned} x_{UB} &= [y_{ub} \;\; v_{ub} \;\; \zeta_{ub}]^T \\ x_{LB} &= [y_{lb} \;\; v_{lb} \;\; \zeta_{lb}]^T. \end{aligned}$$

Controller designs are limited to the region of validity Ω_v. There are no need to presents global stabilization results since they are superfluous due to the physics of the system.

4.2 Reference Trajectory Filter

The reference for the position y is given by a reference trajectory r. The reference trajectory could be a typical clutch sequence, as stated in Løkken (2006):

1. Disengagement of the clutch so that the engine is disconnected from the driveline as fast as possible. The clutch is guaranteed to be disengaged if $r \geq 20mm$. The disengagement is held during the gear shift.
2. Moving the friction plates to the slip point, which is the minimum position between the friction plates without transferring any torque.
3. Slowly moving the friction plates closer until the clutch is completely engaged, referred to as "the critical region".
4. When the clutch is completely engaged and the requested torque is transferred from the engine to the driveline, the desired clutch position is set to $r = 1mm$.

Figure 4.1 shows this typical clutch sequence, which is used in simulations and controller comparisons throughout this thesis.

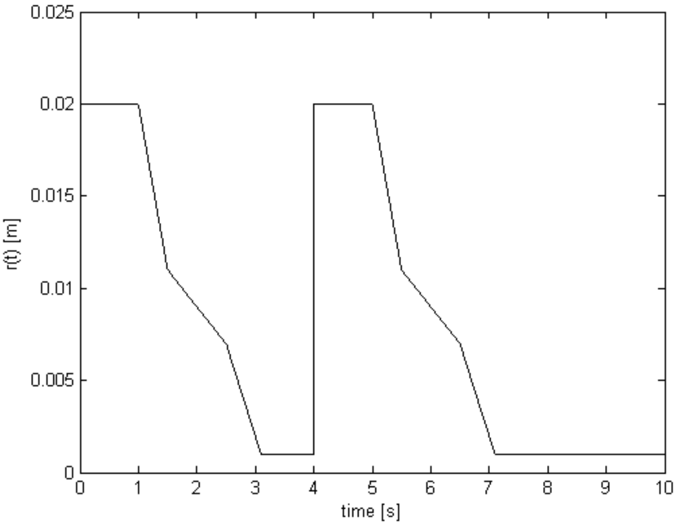


Figure 4.1: A typical clutch sequence

This reference trajectory is filtered when sent to the backstepping-controller, in order to rend a smooth reference signal and derivatives for tracking control. It is an vital assumption for the controller that the reference trajectory and its first η derivatives are continuous and bounded, and available for the controller. Here is η, the relative degree of the backstepping controller, equal to three, thus we need the first three derivatives of the reference trajectory, and four states in our filter to render a smooth signal. The filter is on the form

$$\dot{z} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_0 & -k_1 & -k_2 & -k_3 \end{bmatrix} z + \begin{bmatrix} 0 \\ 0 \\ 0 \\ k_0 \end{bmatrix} r, \qquad z = \begin{bmatrix} y_r \\ \dot{y}_r \\ \ddot{y}_r \\ y_r^{(3)} \end{bmatrix}$$

where r is the unfiltered reference trajectory, and y_r is the reference trajectory after the filtering process. This filter has the characteristic polynomial

$$q = s^4 + k_3 s^3 + k_2 s^2 + k_1 s + k_0.$$

To tune the filter, all the eigenvalues are placed at $\lambda_i = \lambda_\omega$ for $i = 0, 1, 2, 3$, which gives

$$(s + \lambda_\omega)^4 = s^4 + 4s^3\lambda_\omega + 6s^2\lambda_\omega^2 + 4s\lambda_\omega^3 + \lambda_\omega^4$$

Comparing this with the characteristic polynomial of the filter gives the filter coefficients

$$k_0 = \lambda_\omega^4, \qquad k_1 = 4\lambda_\omega^3, \qquad k_2 = 6\lambda_\omega^2, \qquad k_3 = 4\lambda_\omega.$$

This makes the filter critically damped with all the eigenvaules at $-\lambda_\omega$, and with time constant $\tau_t = \frac{4}{\lambda_\omega}$. Experiments using the clutch actuator system in heavy duty trucks have shown that a time constant of $10ms$ is as slow as possible without introducing a noticeable time delay for the driver according to Kaasa (2003), hence $\lambda_\omega = 400$ is used for the third order model.

The reference filter is Hurwitz, and thus exponentially stable, which gives several advantages:

- The tracking reference trajectory y_r becomes smooth.
- Noise and discontinuities in the reference r are filtered out.
- The derivatives of y_r become available as states in $\boldsymbol{z}$, as required by the backstepping-controller.

4.3 Continuous-Time Backstepping

In this section we present the design of a continuous-time backstepping controller for the third-order model of the clutch actuator, based on the design in Løkken (2006). Using backstepping as our control method is valid since (4.1) is on feedback form.

The objectives of the controller design are

- to keep the system state x bounded.
- to achieve tracking; $\lim_{t\to\infty} y(t) = y_r(t)$.

Step one

Define e_1 as the first error variable

$$\begin{aligned} e_1 &= y - y_r \\ \dot{e}_1 &= \dot{y} - \dot{y}_r = v - \dot{y}_r. \end{aligned}$$

Introduce

$$e_2 + \alpha_1 = v - \dot{y}_r$$

The first Lyapunov function is chosen as $V_1 = \frac{1}{2}e_1^2$, and the derivative is

$$\dot{V}_1 = e_1\dot{e}_1 = e_1(v - \dot{y}_r) = e_1(e_2 + \alpha_1) = -c_1e_1^2 + e_1e_2,$$

by choosing the first virtual control as

$$\alpha_1 = -c_1e_1 = -c_1y + c_1y_r.$$

Step two

The derivative of the second error variable is

$$M\dot{e}_2 = M\dot{v} - M\ddot{y}_r - M\dot{\alpha}_1 = -f_l(y) - f_f(v) + Ap(y,\zeta) - AP_0 - M\dot{\alpha}_1 - M\ddot{y}_r.$$

Introducing

$$e_3 + \alpha_2 = Ap(y,\zeta) - M\ddot{y}_r.$$

Choosing the second Lyapunov function as $V_2 = V_1 + \frac{M}{2}e_2^2$ yields

$$\begin{aligned}\dot{V}_2 &= -c_1e_1^2 + e_2(e_1 - f_l(y) - f_f(v) - AP_0 - M\dot{\alpha}_1 + e_3 + \alpha_2)\\ &= -c_1e_1^2 - c_2e_2^2 + e_2e_3\end{aligned}$$

when the second virtual control law is chosen as

$$\begin{aligned}\alpha_2 &= -e_1 - c_2e_2 + f_l(y) + f_f(v) + AP_0 + M\dot{\alpha}_1\\ &= -e_1 - c_2e_2 + f_l(y) + f_f(v) + AP_0 + M\left[\frac{\partial\alpha_1}{\partial y}\dot{y} + \frac{\partial\alpha_1}{\partial y_r}\dot{y}_r\right]\\ &= K_1y + K_2v + K_3y_r + K_4\dot{y}_r + f_l(y) + f_f(v) + AP_0,\end{aligned}$$

where

$$K_1 = -1 - c_1c_2, \qquad K_2 = -c_2 - c_1M, \qquad K_3 = 1 + c_1c_2, \qquad K_4 = c_2 + c_1M.$$

Step three

The derivative of the third error variable is

$$\begin{aligned}\dot{e}_3 &= A\dot{p}(y,\zeta) - My_r^{(3)} - \dot{\alpha}_2\\ &= \frac{A}{V(y)}RT_0\omega - \frac{A^2}{V^2(y)}\zeta v - My_r^{(3)} - \dot{\alpha}_2\end{aligned}$$

Choosing the Lyapunov function $V_3 = V_2 + \frac{1}{2}e_3^2$ yields

$$\begin{aligned}\dot{V}_3 &= -c_1e_1^2 - c_2e_2^2 + e_3\left(e_2 + \frac{A}{V(y)}RT_0\omega - \frac{A^2}{V^2(y)}\zeta v - My_r^{(3)} - \dot{\alpha}_2\right)\\ &= -c_1e_1^2 - c_2e_2^2 - c_3e_3^2 = -\sum_{i=1}^{3}c_ie_i^2 \le 0 \quad \forall c_i > 0\end{aligned}$$

when the control law is chosen as

$$\begin{aligned}\alpha_3 = \frac{A}{V(y)}RT_0\omega &= \frac{A^2}{V^2(y)}\zeta v + My_r^{(3)} - e_2 - c_3e_3 + \dot{\alpha}_2\\ &= K_5y + K_6v + \frac{\partial f_l(y)}{\partial y}v + K_7y_r + K_8\dot{y}_r + K_9\ddot{y}_r + My_r^{(3)} + c_3AP_0 - c_3Ap(y,\zeta)\\ &+ c_3f_l(y) + c_3f_f(v) + \frac{A^2}{V^2(y)}\zeta v + \frac{1}{M}\left[K_2 + \frac{\partial f_f(v)}{\partial v}\right](-f_l(y) - f_f(v) + A(p(y,\zeta) - P_0)),\end{aligned}$$

where

$$K_5 = -c_1 + c_3 K_1, \quad K_6 = -1 + c_3 K_2 + K_1, \quad K_7 = c_1 + c_3 K_3$$
$$K_8 = 1 + c_3 K_4 + K_3, \quad K_9 = c_3 M + K_4.$$

Solving for ω provides the final continuous-time backstepping control law for the third order model of the clutch actuator

$$\omega_{ct} = \frac{V(y)}{ART_0}\alpha_3, \tag{4.2}$$

which is the basis both for the discrete-time controllers presented in the next section, as well as the controllers designed for the fifth order model in the next chapter.

4.3.1 Stability Properties

The Lyapunov function V_3 is positive definite and radially unbounded $\forall e_i \neq 0, \ i = 1, 2, 3$. The derivative $\dot{V}_3$ is negative definite $\forall e_i \neq 0, \ i = 1, 2, 3$. Hence, V is non-increasing and converges as $t \to \infty$. V is bounded below by zero, and the error variables e_i are bounded. Since the reference trajectory y_r and its derivatives are bounded by assumption, the system states are bounded. It can be shown, using the Comparison Lemma (see Appendix A) that the error dynamics converge to zero exponentially, thus $\lim_{t\to\infty} y(t) = y_r(t)$. Hence, the controller achieves:

- exponential tracking for the position y in the region of validity Ω_v.
- all system states remain bounded.

The design parameters available for the controller are $c_i > 0$ for $i = 1, 2, 3$.

4.3.2 Simulation

It should be mentioned that all simulations in this thesis are conducted with full state feedback from the system plant, as opposed to in practice, where the clutch actuator uses output feedback with only measurement of the position, and a nonlinear observer to estimate the remaining states. This thesis is concentrated on controller comparisons for sampled-data systems, thus not including output feedback design.

The continuous-time controller is simulated[1] to prove good tracking performance of the typical clutch sequence mentioned above. Figure 4.2 shows the trajectory tracking response of the system in an ideal simulation, where there are no bounds on the system states. Looking at the states and input in figure 4.3, we clearly observe that the velocity and specially the mass flow input is unrealistic high compared to the bounds stated in the region of validity. Setting bounds on the states and input is necessary to preserve the physics of the actuator, and will result in a slower response of the system.

[1] All simulations in this thesis is done with a Dormand-Price simulation solver with a fixed step size of 2ms.

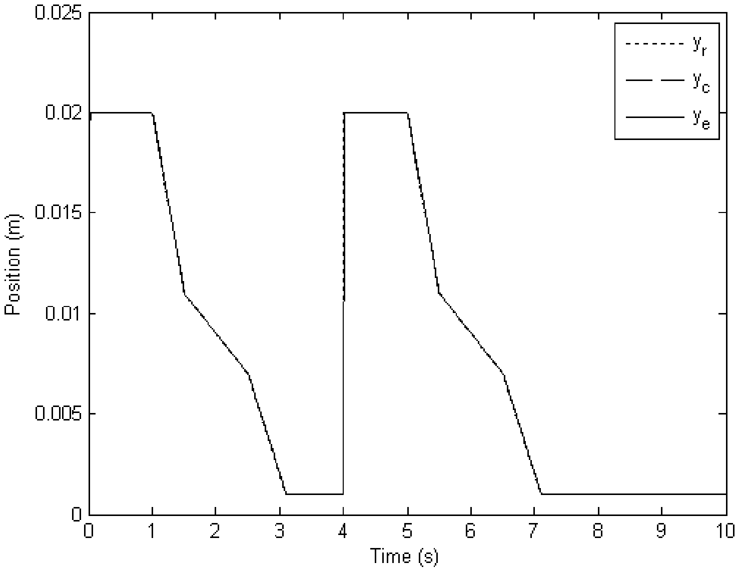


Figure 4.2: Simulation of the continuous-time controller without bounds

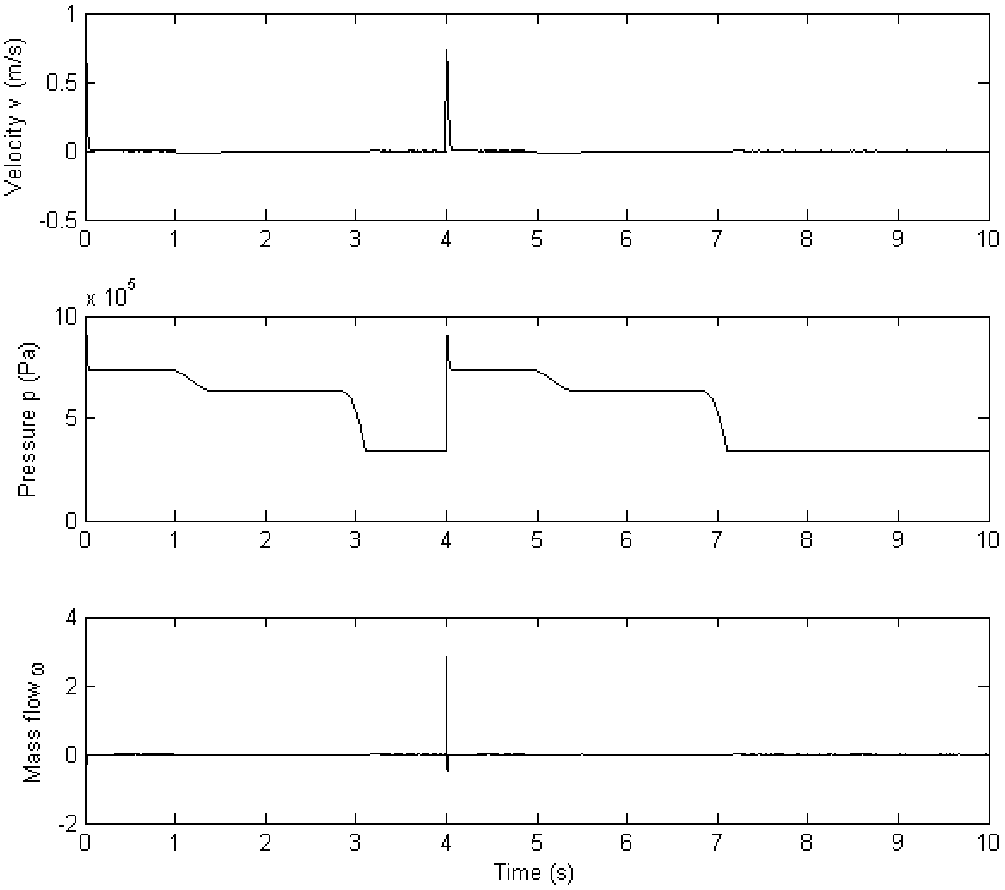


Figure 4.3: Simulated system states and inputs without bounds

Introducing the region of validity to our simulation system gives a more realistic simulation, as presented in figure 4.4, where we can observe the system responding somewhat slower when the reference trajectory changes rapidly. From the zoomed-in plot we see that the reference trajectory filter renders a smooth trajectory y_r from the unfiltered reference r, and futhermore we see the measured position y_c closely following y_r, albeit with time deviation when there are sudden movements on the clutch.

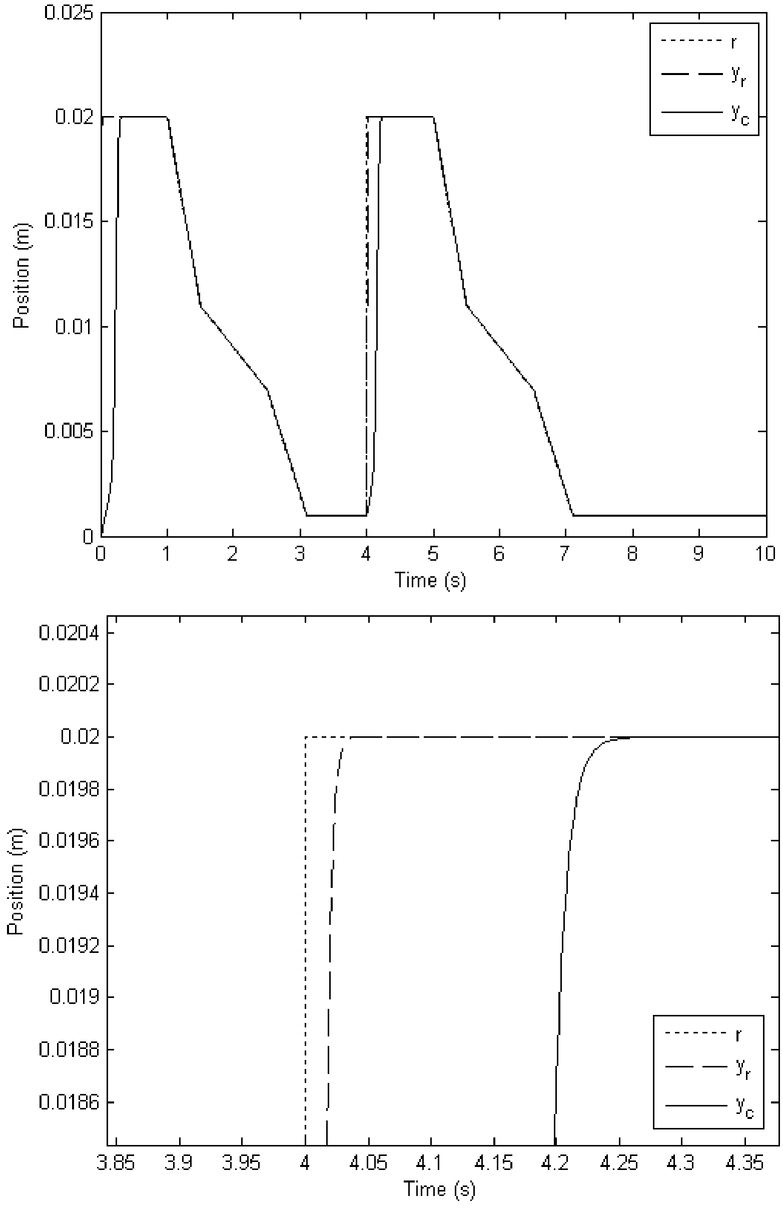


Figure 4.4: Simulation of the continuous-time controller with bounds

For the continuous-time backstepping controller the design constants are chosen as $c_i = 1000,\ i = 1, 2, 3$.

Taking another look at the system states and input as in figure 4.5 we see the system is now operating well within it's physical limits.

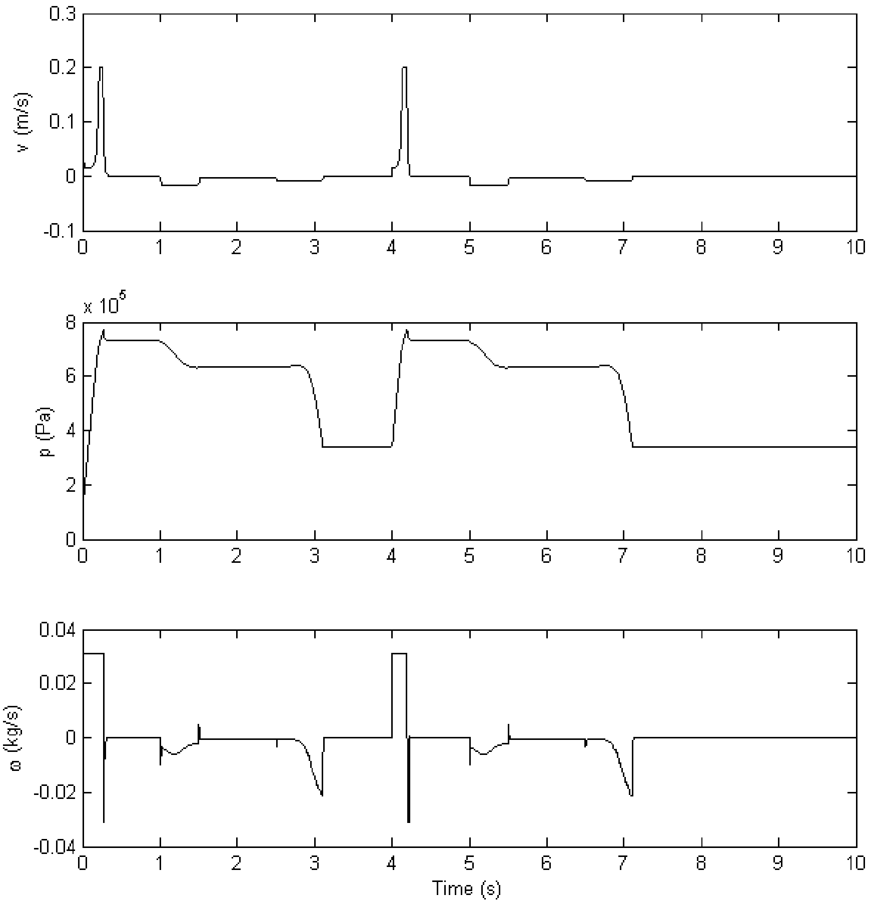


Figure 4.5: Simulated system states and inputs with bounds

To conclude, the continuous-time backstepping design clearly renders a controller with good trajectory tracking performance. How the digital implementation of the controller changes the conditions for the system will be shown in the controller comparison later in this chapter. But first, we design the discrete-time controllers.

4.4 Discrete-Time Backstepping

In this section the two methods for backstepping in the discrete-time domain are presented.

4.4.1 Emulation Design

From Subsection 2.1.1 we know that emulation design in practice is just a discretization of the already obtained continuous-time controller, with *sampler* and *zero-order hold* devices before and after the controller, respectively. In this case, the emulation controller is

$$\omega_{em} = \omega_{ct}(k), \tag{4.3}$$

that achieves global asymptotic stability of the closed loop system (4.1), (4.3) according to the general and unified framework presented in Laila (2003). Next, we design a controller using the direct discrete-time design.

4.4.2 Direct Discrete-Time Design

The direct discrete-time design of the controller is based on the Euler approximate model, following the steps in Chapter 3. The third order Euler approximate model is

$$\begin{aligned} y(k+1) &= y(k) + Tv(k) \\ Mv(k+1) &= Mv(k) + T(-f_l(y(k)) - f_f(v(k)) - AP_0 + Ap(k)) \\ \zeta(k+1) &= \zeta(k) + TRT_0\omega. \end{aligned} \tag{4.4}$$

For the sake of simplicity we omitt the argument (k) in the rest of this section: it should be clear by now that we operate in the discrete-time domain.

The parameters used in the design model are obtained from the second step of the continuous-time backstepping procedure of the previous section

$$\begin{aligned} \alpha_T &= K_1 y + K_2 v + K_3 y_r + K_4 \dot{y}_r + M\ddot{y}_r + f_l(y) + f_f(v) + AP_0 \\ W_T &= \frac{1}{2}(y - y_r)^2 + \frac{M}{2}(v - \dot{y}_r - c_1 y_r + c_1 y)^2. \end{aligned}$$

First step of the design is to calculate $\Delta\alpha_T$;

$$\begin{aligned} \Delta\alpha_T &= K_1(y + Tv) + K_2(v + T\dot{v}) + K_3(y_r + T\dot{y}_r) + K_4(\dot{y}_r + T\ddot{y}_r) + M(\ddot{y}_r + Ty_r^{(3)}) + f_l(y + Tv) \\ &\quad + f_f(v + T\dot{v}) + AP_0 - K_1 y + K_2 v + K_3 y_r + K_4 \dot{y}_r + M\ddot{y}_r - f_l(y) - f_f(v) - AP_0 \\ &= T\left(\left(K_1 + \frac{\partial f_l(y)}{\partial y}\right)v + \left[K_2 + \frac{\partial f_f(v)}{\partial v}\right]\dot{v} + K_3\dot{y}_r + K_4\ddot{y}_r + My_r^{(3)}\right). \end{aligned}$$

In the above equation, the Mean Value Theorem (see Appendix A) has been used, and in reality it is

$$\frac{\partial f_l(y')}{\partial y'},\ y' \in [y, Tv] \quad and \quad \frac{\partial f_f(v')}{\partial v'},\ v' \in [v, T\dot{v}].$$

Next, $\overline{\Delta W}_T$ is calculated (using $(a^2 - b^2) = (a+b)(a-b)$);

$$\begin{aligned} \overline{\Delta W}_T &= \frac{M}{2}\left(\left(v + \frac{T}{M}(-f_l(y) - f_f(v) - AP_0 - Ap)\right) + c_1(y + Tv) - (\dot{y}_r + T\ddot{y}_r) - c_1(y_r + T\dot{y}_r)\right)^2 \\ &\quad - \frac{M}{2}\left(\left(v + \frac{T}{M}(-f_l(y) - f_f(v) - AP_0 - \alpha_T)\right) + c_1(y + Tv) - (\dot{y}_r + T\ddot{y}_r) - c_1(y_r + T\dot{y}_r)\right)^2 \\ &= \left[v + c_1 y - \dot{y}_r - c_1 y_r + \frac{T}{M}\left(-f_l(y) - f_f(v) - AP_0 + \frac{1}{2}(Ap - \alpha_T) - v - \ddot{y}_r - \dot{y}_r\right)\right] \cdot T(Ap - \alpha_T), \end{aligned}$$

which gives

$$\widetilde{\Delta W}_T = T\left(v - \dot{y}_r - c_1 y_r + c_1 y + \frac{T}{2M}(K_1 y + K_2' v - f_l(y) - f_f(v) - AP_0 + Ap + K_3 y_r + K_4' \dot{y}_r - M\ddot{y}_r)\right)$$

where $K_2' = -1 - c_1 M$ and $K_4' = c_2 - c_1 M$.

Finally, the control law can be written

$$\begin{aligned}
u &= c_3(\alpha_T - Ap) - \frac{\widetilde{\Delta W}_T}{T} + \frac{\Delta \alpha_T}{T} \\
&= K_5 y + K_6 v + \frac{\partial f_l(y)}{\partial y} v + c_3(AP_0 - Ap + f_l(y) + f_f(v)) + K_7 y_r \\
&+ K_8 \dot{y}_r + K_9 \ddot{y}_r + M y_r^{(3)} + \frac{1}{M}\left(K_2 + \frac{\partial f_f(v)}{\partial v}\right)(-f_l(y) - f_f(v) - AP_0 + Ap) \\
&- \frac{T}{2M}\left(K_1 y + K_2' v - f_l(y) - f_f(v) - AP_0 + Ap + K_3 y_r + K_4' \dot{y}_r - M\ddot{y}_r\right),
\end{aligned}$$

which, per definition, leads to

$$u = A\dot{p} = \frac{A}{V} R T_0 \omega - \frac{A^2}{V^2}\zeta v \qquad \Rightarrow \qquad \omega = \frac{V}{ART_0}\left(u + \frac{A^2}{V^2}\zeta v\right).$$

This yields the final control law

$$\omega_{dt} = \omega_{ct} - \frac{VT}{2MART_0}\left(K_1 y + K_2' v - f_l(y) - f_f(v) - AP_0 + Ap + K_3 y_r + K_4' \dot{y}_r - M\ddot{y}_r\right), \quad (4.5)$$

which achieves semiglobal-practical stability of the approximate discrete-time control system (4.4), (4.5), and furthermore asymptotic stability of the closed loop sampled-data system (4.1), (4.5) in the region of validity Ω_v. This can be proved with the Lyapunov function $V_T = W_T + \frac{1}{2}(Ap - \alpha_T)^2$.

4.5 Controller Comparison

This section contains simulation results of the 3 presented controllers (4.2), (4.3) and (4.5). The controllers are simulated for different sampling intervals, as presented below. The output from the continuous-time system y_c is used as a reference for the outputs of the sampled-data systems, since theoretically the sampled-data systems can never perform better than the pure continuous-time system. We do not include y_r in the plots, since the deviation between y_r and y_c has already been established, and naturally it stays the same for all sampling intervals.

The controllers are expected to behave in a similar fashion for low sampling intervals, thus the aim of this section is to discover whether the system with the direct discrete-time designed controller perform better than the system with the emulation-controller when raising the sampling interval.

4.5.1 Simulation with T = 2ms

We start this controller comparison with a sampling interval of 2ms, which is the same as the fixed step size used in the simulation solver. For all the simulations in this thesis, y_e means the output from the sampled-data system with emulation controller, and y_d means the output from the system with the direct discrete-time designed controller. From figure 4.6 we see that the two discrete-time controllers perfom well with this sampling interval, which is expected.

In order too on conclude the performance of the controllers, we also look at the tracking error compared to the reference trajectory y_r. As shown in figure 4.7, it is not possible to separate the two discrete-time controllers in terms of trajectory tracking with this sampling interval. For $T = 2ms$, the backstepping constants are set to $c_1 = c_2 = c_3 = 600$ for both controllers.

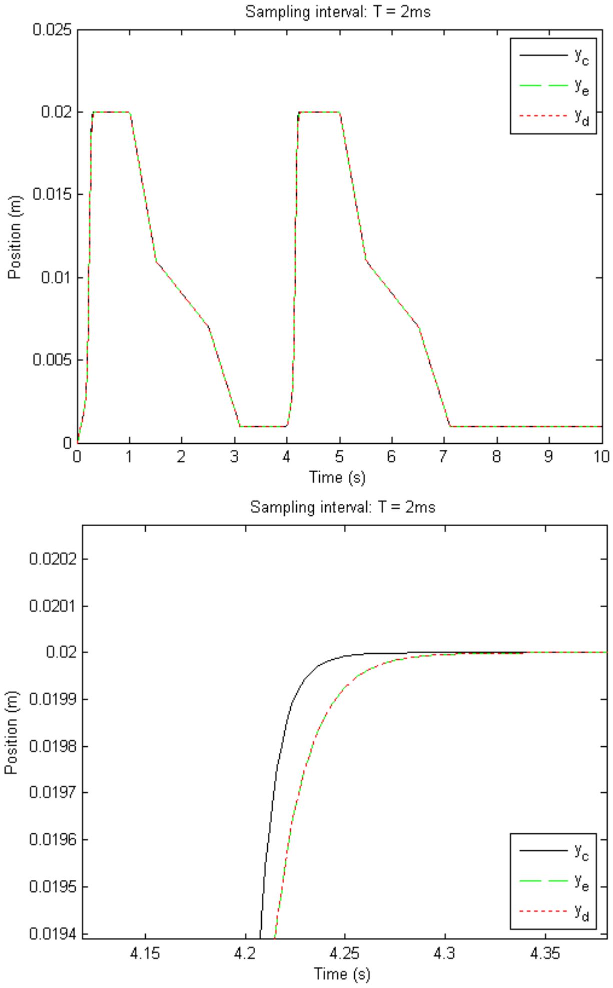


Figure 4.6: Controller comparison, T = 2ms

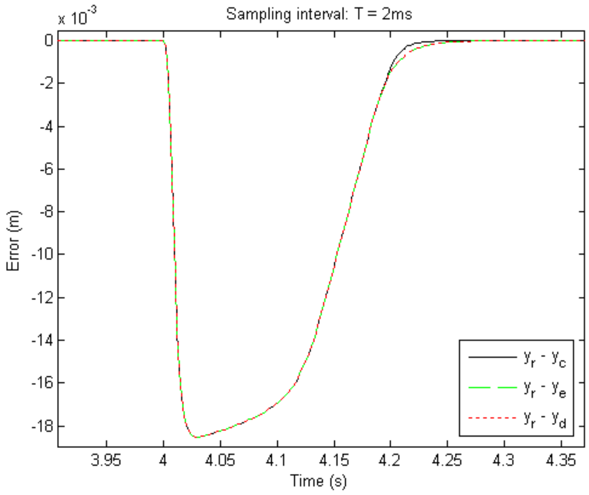


Figure 4.7: Trajectory tracking error, T = 2ms

4.5.2 Simulation with T = 6ms

Next, we raise the sampling interval to $T = 6ms$, which is three times the step size of the solver, but still slower than the currently used sampling interval of the clutch actuator. From figure 4.8 it is obvious that the two discrete-time controllers still give very good trajectory tracking, although not as smooth as for the $2ms$ case.

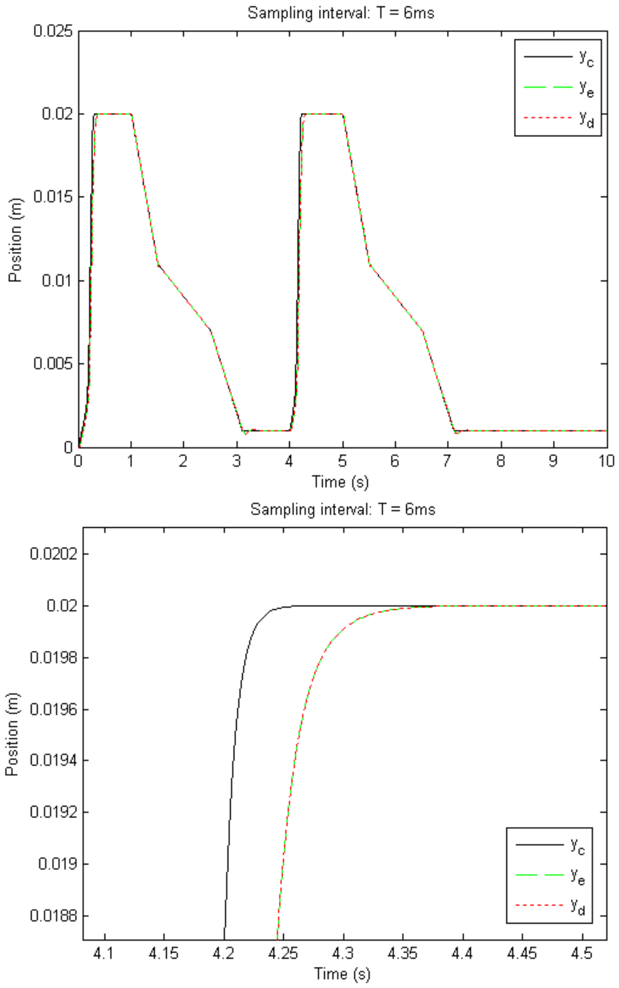


Figure 4.8: Controller comparison, T = 6ms

From the plots we see that there is no noticeable difference in the performance of the two controllers for this sampling interval. For $T = 6ms$, the backstepping constants are chosen as $c_1 = c_2 = 600,\ c_3 = 40$ for both controllers.

4.5.3 Simulation with T = 10ms

A sampling interval of $T = 10ms$ is perhaps the most interesting, since this is the sampling currently in use by KA on their clutch actuator. As we can see from figure 4.9, this sampling interval yields some overshoot and small oscillations due to the slower reaction of the system. Still, however, the controllers give good position tracking, and the overshoot is not really of any concern due to the mechanics of the system that ensures the clutch is disengaged as long as $y \geq 20mm$. For this sampling interval, the backstepping constants are chosen to be $c_1 = c_2 = 400,\ c_3 = 25$.

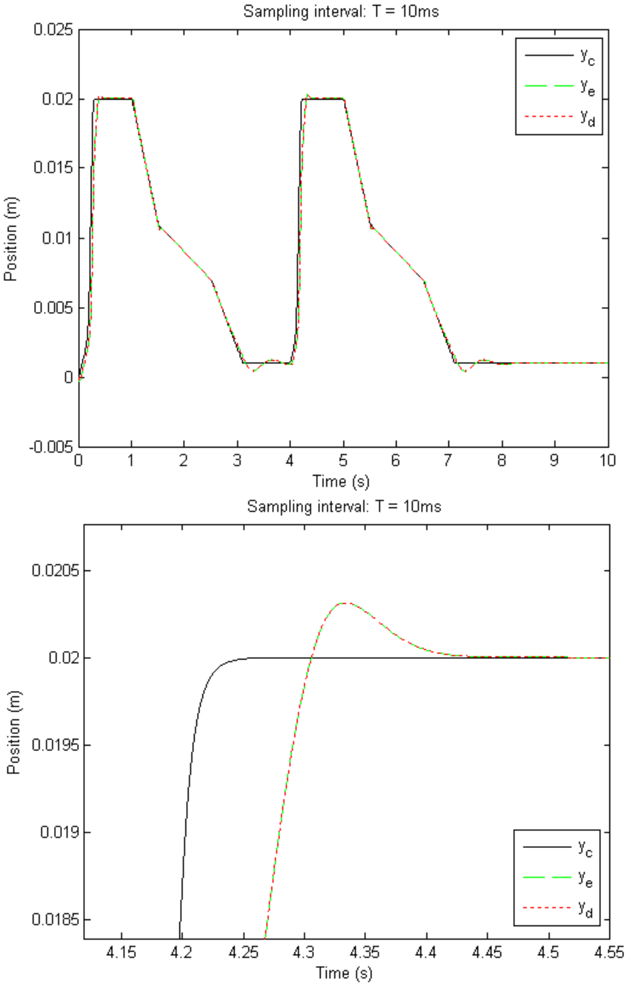


Figure 4.9: Controller comparison, T = 10ms

At this point it is clear that the digital implementation of the continuous-time controller *does* lower the trajectory tracking performance of the sampled-data system, as opposed to an analog implementation. Although, since this is a simulation with ideal conditions, the performance of the sampled-data systems are still very decent. In a practical attempt there might be unmodelled dynamics and measurement noise that could further decrease the performance of the sampled-data systems.

It should be mentioned that all possible sets of backstepping constants have not been tried for all the different sampling intervals. This is due to the time-consumation this would take, and the fact that the goal of this thesis is not to tune the controllers the best way possible, but to investigate differences in performance of the two discrete-time controllers.

However, for this sampling interval there is still nothing to separate the two controllers performancewise, as we can see from the tracking error in figure 4.10. On the tracking error plots, the part that is below zero is because of slow reaction due to bounds on system states and inputs, and the part that is above zero is mostly overshoot due to the decreasingly slower sampling.

Looking at the states and system input for this sampling rate, we see that the states and input are well within the defined region of validity. Figure 4.11 shows states and input of the sampled-data system with the direct discrete-time controller.

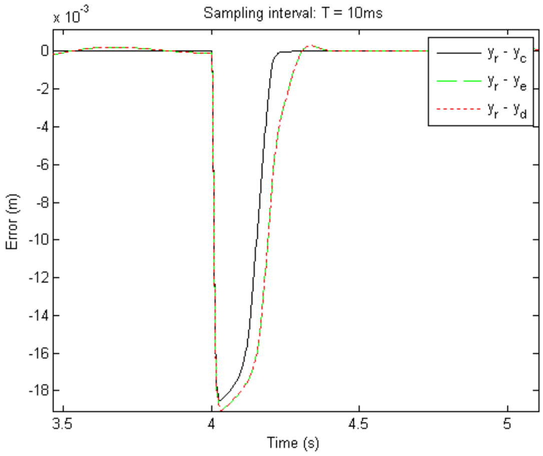


Figure 4.10: Trajectory tracking error, T = 10ms

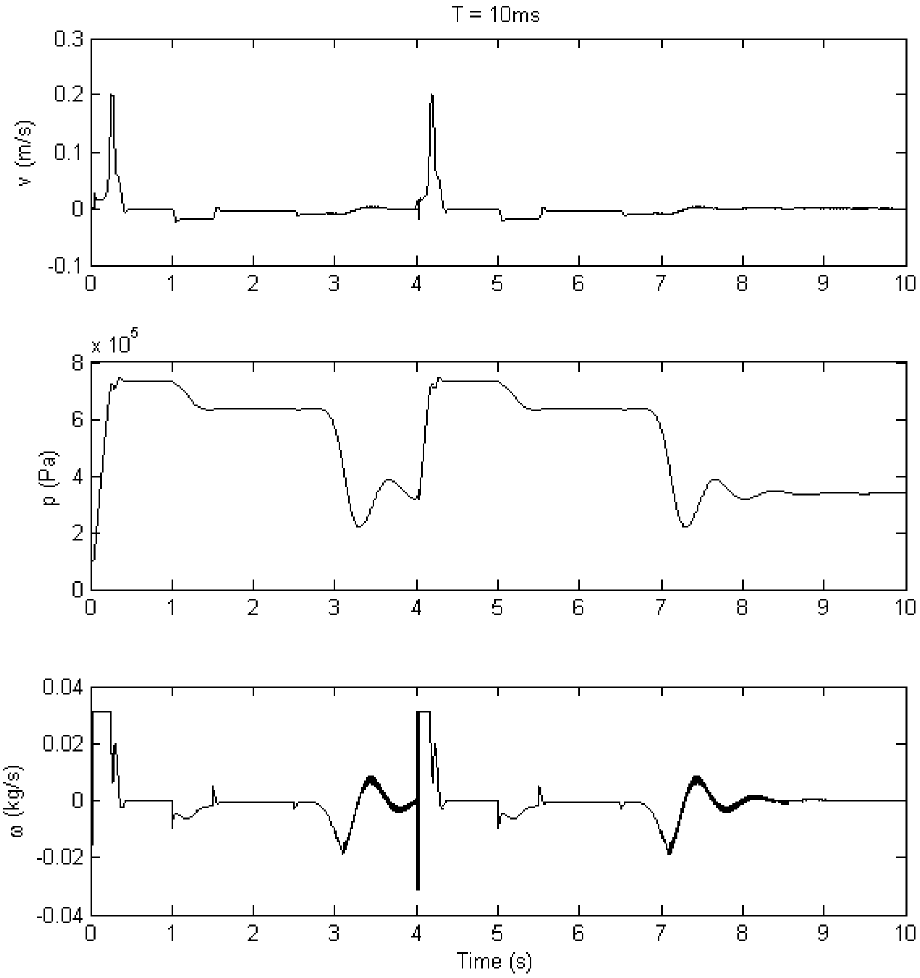


Figure 4.11: Simulated system states and inputs, T = 10ms

4.5.4 Simulation with T = 16ms

When choosing a higher sampling interval like $T = 16ms$, as in figure 4.12, we still observe good tracking performance with both controller. Eveh with some more overshoot and damped oscillations than before, it seems like this sampling interval could be tried in a practical implementation of the system. But, as we see, there are still no difference in the performance of the two discrete-time controllers. For this sampling rate, the design constants are set to $c_1 = c_2 = 400$, $c_3 = 20$. The third design constant needs to be chosen lower than the other two, since this one corresponds to the pressure which has a lot higher value then the position and velocity.

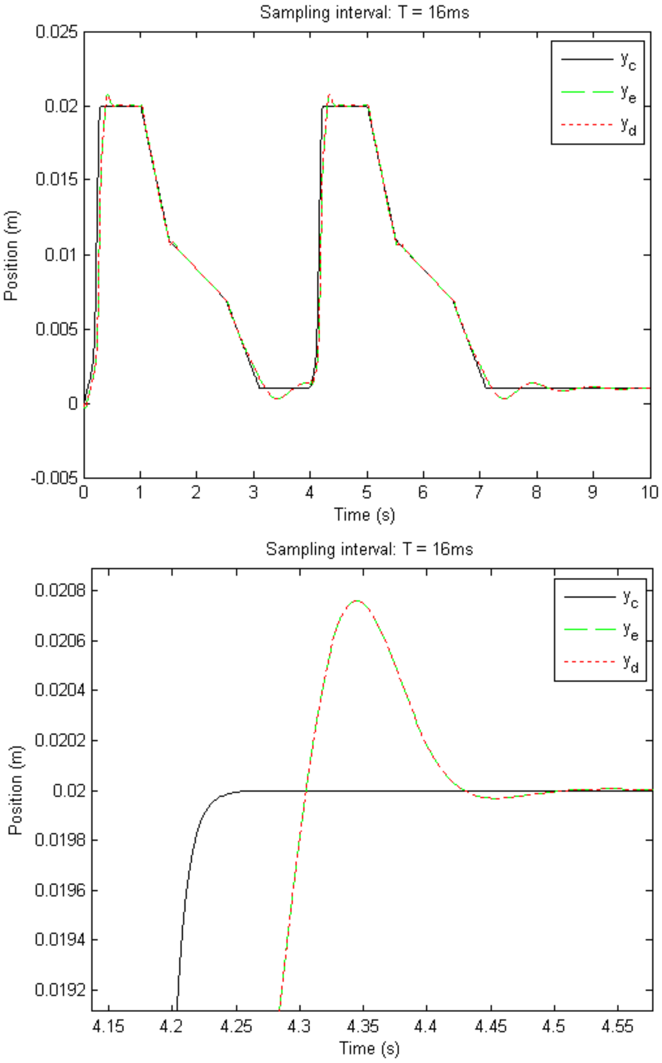


Figure 4.12: Controller comparison, T = 16ms

To see if we can find some difference in the controllers at all, we move to an even higher sampling interval. Higher sampling intervals are where we would expect the direct discrete-time designed controller to really outperform the emulation controller, since the sampling feature of the systems gets even more crucial for their operation.

4.5.5 Simulation with T = 30ms

With a higher sampling interval of $T = 30ms$, which is three times slower sampling than the clutch actuator currently operates within, the position tracking of the sampled-data systems clearly is to slow and vulnerable: Although, some sensible trajectory tracking is still preserved as shown in figure 4.13 and 4.14. Both systems are still stable, since the constraints on the positions are not violated. The backstepping constants are here set to $c_1 = c_2 = 400,\ c_3 = 15$.

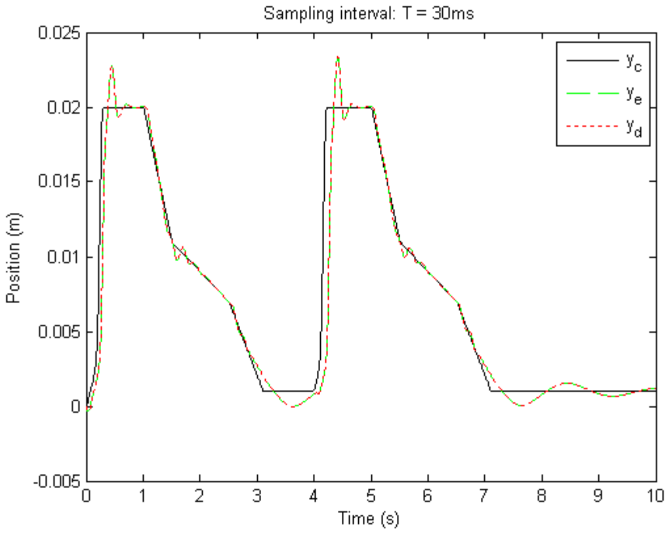


Figure 4.13: Controller comparison, T = 30ms

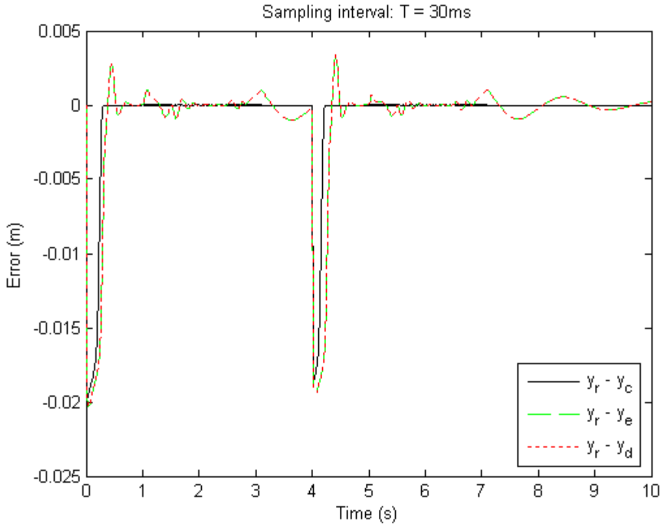


Figure 4.14: Trajectory tracking error, T = 30ms

Surprisingly enough, there is still no difference in tracking performance of the two discrete-time controllers even for this higher sampling interval, which suggests that the controller designed when taking account of the sampling have not reached the potential shown earlier in the thesis. Why this is will be shown in the conclusion in the next section.

4.6 Conclusion

As shown in the controller comparison in the last section, there is no observable difference in trajectory tracking performance of the two discrete-time controllers. Figure 4.15 shows the contributions from the continuous-time and discrete-time terms, before saturation, in the direct discrete-time controller when using a sampling interval of $10ms$, zoomed in around $t = 4s$ to give a somewhat readable graph. As shown, the discrete-time terms is neglectable for the low sampling intervals that the clutch actuator operates within, thus loosing whatever properties of improvement the controller had compared to the emulated controller.

Why the discrete-time terms are negligible may be a coincidence, or it might be that our system is somehow not suited for this design. Indeed, in all the examples in the literature, the design is applied to somewhat less complex nonlinear systems. Since higher sampling interval gives higher value of the discrete-time terms, improvements might occur for sampling intervals too high to be used in practice, e.g. really high sampling intervals of some hundred milliseconds as shown in the case study in Section 2.2.

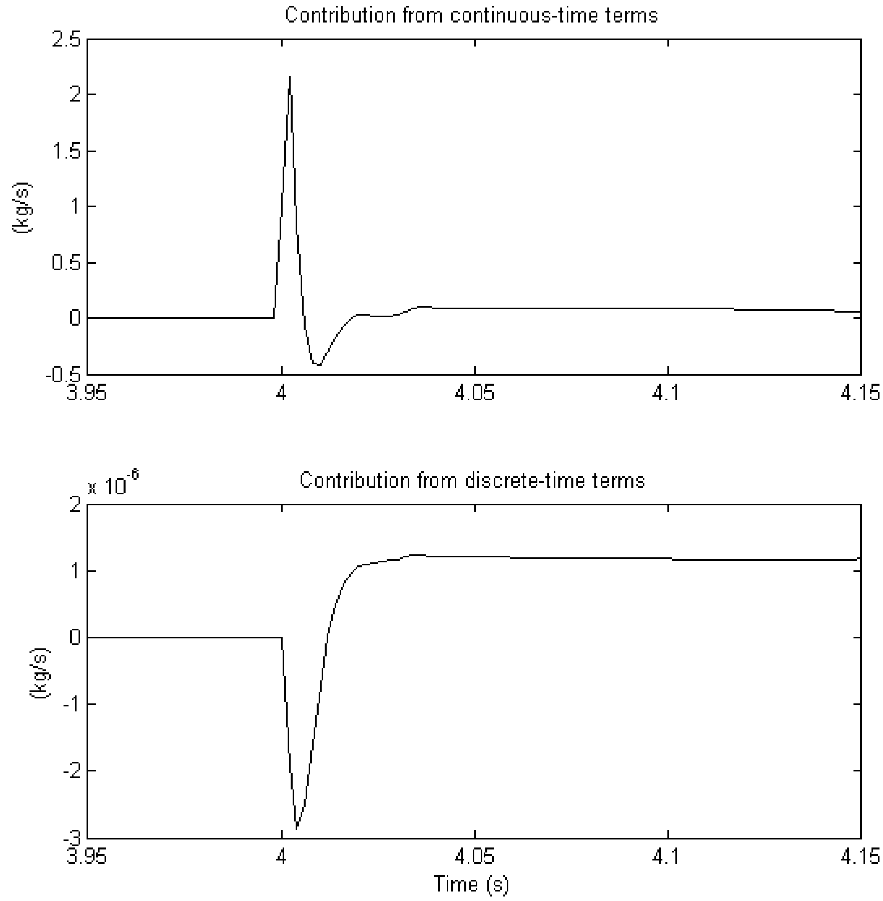


Figure 4.15: Contributions from continuous- and discrete-time terms in controller u_{dt}

Since we want to observe and compare the controllers for the most realistic model possible, we also want to try our design on the fifth order model of the clutch actuator. For now, we merely conclude that the discrete-time design does not *seem* to render a controller with improved trajectory tracking properties for our system. We move on to design and investigate controllers for the fifth order model, and leave the final conclusion to the following chapters.

Chapter 5

5th Order Model

This chapter contains a fifth order design model presented in Løkken (2006), together with both continuous-time and discrete-time controller designs, and comparison between these. The fifth order model include the entire dynamics of the physical system described in Section 1.1, with the beforementioned assumptions of isothermal conditions and constant pressure in chamber B.

5.1 Mathematical Model

The fifth order model of the electro-pneumatic actuator is mathematically modelled as

$$\begin{aligned}
\dot{y} &= v \\
M\dot{v} &= -f_l(y) - f_f(v, y_f) + A\left[p(y,\zeta) - P_0\right] \\
\dot{\zeta} &= RT_0\omega \\
\tau_v\dot{\omega} &= -\omega + q_v(p(y,\zeta), u_v) \\
\dot{y}_f &= g_f(v, y_f),
\end{aligned} \tag{5.1}$$

where the first three equations is as described in Chapter 4, except the introduction of the Lugre friction model that is given by

$$\begin{aligned}
f_f(v, y_f) &= D_v v + K_f y_f + D_f g_f(v, y_f) \\
g_f(v, y_f) &= v - \frac{K_f}{F_d}|v|_s y_f,
\end{aligned}$$

where D_v, K_f and F_d are friction coefficients, and y_f describes the pre-sliding deflection of the objects in contact. It is not a real measureable state, and is only included to model a spring-like behaviour of the objects before they slip. It enters the system as a kind of internal dynamics, and is given in order to render a more accurate friction force for inclusion of friction compensation in the controllers. The absolute value in the Lugre model is replaced with a smoothed approximation

$$|v| \approx |v|_s = \sqrt{v^2 + \epsilon^2}$$

where $\epsilon > 0$ is an arbitrary small parameter. The flow through the valve is given by

$$q_v(p(y,\zeta), u_v) = \rho_0 C_v \left[\psi\left(\frac{p}{P_S}\right) P_S \cdot h_v(u_v) - \psi\left(\frac{P_E}{p}\right) p \cdot h_v(-u_v)\right],$$

where P_S and P_E is the supply and exhaust pressure, respectively.

The pressure ratio function is chosen to be

$$\psi(r) = \sqrt{1 - r^2}, \qquad r \in [0, 1],$$

and the function h_v can be modelled simply as

$$h_v(u_v) = \begin{cases} u_v, & u_v > 0 \\ 0, & u_v \leq 0 \end{cases},$$

where u_v is the actual control input.

All the parameters for the fifth order model in table 5.1 are obtained from Løkken (2006).

Parameter	Value	Unit
M	10	kg
A	0.0123	m^2
P_0	$1 \cdot 10^5$	Pa
P_E	$1 \cdot 10^5$	Pa
P_S	$9 \cdot 10^5$	Pa
R	288	J/(kgK)
T_0	293	K
V_0	0,0008	m^3
D_v	2071,9	Ns/m
K_f	46595	N/m
D_f	100	Ns/m
F_d	209,99	N
ρ_0	1,185	kg/m^3
C_v	$2,8950 \cdot 10^{-8}$	$m^3/(Pa \cdot s)$
τ_v	0,0344	1/s

Table 5.1: Parameters used for the fifth order model

In this model the valve dynamics and an imitated time constant are combined into a single dynamical description to simplify the model. The time constant is introduced to imitate the time bias appearing due to the use of on/off valves instead of proportional valves. No leakage in the valve is assumed.

5.1.1 Region of Validity

In Subsection 4.1.1 the physical limits on the first four state variables are presented. In addition, y_f is limited by $|y_f| \leq \frac{F_d}{K_f}$, and there are limits due to the construction of valve electronics $u_{v,lb} = -1V$ and $u_{v,ub} = 1V$. The feasible region for the system states is then

$$\Omega_v = \left\{\forall x, \forall u_v | x_{LB} \leq x \leq x_{UB} | u_{v,lb} \leq u_v \leq u_{v,ub}\right\},$$

where the upper and lower boundaries are defined as

$$\begin{aligned} x_{UB} &= \left[y_{ub} \quad v_{ub} \quad \zeta_{ub} \quad \omega_{ub} \quad y_{f,ub}\right]^T \\ x_{LB} &= \left[y_{lb} \quad v_{lb} \quad \zeta_{lb} \quad \omega_{lb} \quad y_{f,lb}\right]^T. \end{aligned}$$

Controller designs are limited to the region of validity Ω_v. There are no need to presents global stabilization results since they are superfluous due to the physics of the system.

5.2 Reference Trajectory Filter

The reference trajectory filter for the fifth order model is made for the same reasons, and based on the same method, as for the third order model in Section 4.2. The filter is on the form

$$\dot{z} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -k_0 & -k_1 & -k_2 & -k_3 & -k_4 \end{bmatrix} z + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ k_0 \end{bmatrix} r, \qquad z = \begin{bmatrix} y_r \\ \dot{y}_r \\ \ddot{y}_r \\ y_r^{(3)} \\ y_r^{(4)} \end{bmatrix}$$

where r is the unfiltered reference trajectory and y_r is the reference trajectory after the filtering process. The filter has the characteristic polynomial

$$q = s^5 + k_4 s^4 + k_3 s^3 + k_2 s^2 + k_1 s + k_0.$$

To tune the filter, all the eigenvalues are placed at $\lambda_i = \lambda_\omega$ for $i = 0, 1, 2, 3, 4$, which gives

$$(s + \lambda_\omega)^5 = s^5 + 5s^4\lambda_\omega + 10s^3\lambda_\omega^2 + 10s^2\lambda_\omega^3 + 5s + \lambda_\omega^5.$$

Comparing this with the characteristic polynomial of the filter gives the filter coefficients

$$k_0 = \lambda_\omega^5, \qquad k_1 = 5\lambda_\omega^4, \qquad k_2 = 10\lambda_\omega^3, \qquad k_3 = 10\lambda_\omega^2, \qquad k_4 = 5\lambda_\omega.$$

This makes the filter critically damped with all the eigenvaules at $-\lambda_\omega$, and with a time constant $\tau_t = \frac{5}{\lambda_\omega}$. As mentioned, a time constant at $10ms$ is as slow as possible without introducing a noticeable time delay for the driver, hence $\lambda_\omega = 500$ is used for the fifth order model.

The reference filter is Hurwitz, thus obtaining a smooth trajectory y_r for tracking purposes, and making the first four derivatives of y_r available for our controller design.

5.3 Continuous-Time Backstepping

Defining the fourth error variable as

$$e_4 = \frac{A}{V} R T_0 \omega - M y_r^{(3)} - \alpha_3$$

yields

$$\dot{V}_3 = -c_1 e_1^2 - c_2 e_2^2 - c_3 e_3^2 + e_3 e_4$$

when the third virtual control law is chosen as (see Section 4.3)

$$\begin{aligned} \alpha_3 = {} & K_5 y + K_6 v + \frac{\partial f_l(y)}{\partial y} v + K_7 y_r + K_8 \dot{y}_r + K_9 \ddot{y}_r + c_3 A P_0 - c_3 A p(y,\zeta) + c_3 f_l(y) + c_3 f_f(v, y_f) \\ & + \frac{A^2}{V^2} \zeta v + \frac{\partial f_f(v, y_f)}{\partial y_f} g_f(v, y_f) + \frac{1}{M} \left[K_2 + \frac{\partial f_f(v, y_f)}{\partial v} \right] (-f_l(y) - f_f(v, y_f) + A(p(y,\zeta) - P_0)). \end{aligned}$$

Note that the Lugre friction model is accounted for here, as opposed to for the third order model, where the simple viscous friction model was used.

Step four

The derivate of the fourth error variable is

$$\dot{e}_4 = -\frac{A^2}{V^2}RT_0\omega v + \frac{A}{V}RT_0\dot{\omega} - My_r^{(4)} - \dot{\alpha}_3.$$

Choosing the Lyapunov function $V_4 = V_3 + \frac{1}{2}e_4^2$ yields, when derivated

$$\begin{aligned}\dot{V}_4 &= -\sum_{i=1}^{3} c_i e_i^2 + e_4(e_3 - \frac{A^2}{V^2}RT_0\omega v + \frac{ART_0}{\tau_v V}(-\omega + q_v(p(y,\zeta),u_v)) - My_r^{(4)} - \dot{\alpha}_3)\\ &= -\sum_{i=1}^{4} c_i e_i^2 < 0, \qquad \forall c_i > 0,\ \forall e_i \neq 0, i = 1,2,3,4,\end{aligned}$$

when the control law is chosen as

$$\alpha_4 = \frac{ART_0}{\tau_v V}q_v(p(y,\zeta),u_v) = -e_3 - c_4 e_4 + My_r^{(4)} + \frac{ART_0}{V}\left(\frac{A}{V}v + \frac{1}{\tau_v}\right)\omega + \dot{\alpha}_3.$$

Due to the complexity, $\dot{\alpha}_3$ can be evaluated as

$$\dot{\alpha}_3 = \frac{\partial\alpha_3}{\partial y}\dot{y} + \frac{\partial\alpha_3}{\partial v}\dot{v} + \frac{\partial\alpha_3}{\partial\zeta}\dot{\zeta} + \frac{\partial\alpha_3}{\partial y_f}\dot{y}_f + \frac{\partial\alpha_3}{\partial y_r}\dot{y}_r + \frac{\partial\alpha_3}{\partial\dot{y}_r}\ddot{y}_r + \frac{\partial\alpha_3}{\partial\ddot{y}_r}y_r^{(3)},$$

where

$$\begin{aligned}\frac{\partial\alpha_3}{\partial y} &= K_5 + \frac{\partial^2 f_l(y)}{\partial y^2}v + c_3\frac{\partial f_l(y)}{\partial y} + c_3\frac{A^2}{V^2}\zeta - \frac{2A^3}{V^3}\zeta v + \frac{1}{M}\left[K_2 + \frac{\partial f_f(v,y_f)}{\partial v}\right]\left(-\frac{\partial f_l(y)}{\partial y} - \frac{A^2}{V^2}\zeta\right)\\ \frac{\partial\alpha_3}{\partial v} &= K_6 + \frac{\partial f_l(y)}{\partial y} + c_3\frac{\partial f_f(v,y_f)}{\partial v} + \frac{A^2}{V^2}\zeta + \frac{\partial^2 f_f(v,y_f)}{\partial v\partial y_f}g_f(v,y_f) + \frac{\partial f_f(v,y_f)}{\partial y_f}\frac{\partial g_f(v,y_f)}{\partial v}\\ &\quad + \frac{1}{M}\frac{\partial^2 f_f(v,y_f)}{\partial v^2}(-f_l(y) - f_f(v,y_f) + Ap - AP_0) - \frac{1}{M}\left[K_2 + \frac{\partial f_f(v,y_f)}{\partial v}\right]\frac{\partial f_f(v,y_f)}{\partial v}\\ \frac{\partial\alpha_3}{\partial\zeta} &= -c_3\frac{A}{V} + \frac{A^2}{V^2}v + \frac{1}{M}\left[K_2 + \frac{\partial f_f(v,y_f)}{\partial v}\right]\frac{A}{V}\\ \frac{\partial\alpha_3}{\partial y_f} &= c_3\frac{\partial f_f(v,y_f)}{\partial y_f} + \frac{\partial^2 f_f(v,y_f)}{\partial y_f^2}g_f(v,y_f) + \frac{\partial f_f(v,y_f)}{\partial y_f}\frac{\partial g_f(v,y_f)}{\partial y_f}\\ &\quad - \frac{1}{M}\left[K_2 + \frac{\partial f_f(v,y_f)}{\partial v}\right]\frac{\partial f_f(v,y_f)}{\partial y_f} + \frac{1}{M}\frac{\partial^2 f_f(v,y_f)}{\partial y_f\partial v}(-f_l(y) - f_f(v,y_f) + Ap - AP_0)\\ \frac{\partial\alpha_3}{\partial y_r} &= K_7 \qquad \frac{\partial\alpha_3}{\partial\dot{y}_r} = K_8 \qquad \frac{\partial\alpha_3}{\partial\ddot{y}_r} = K_9.\end{aligned}$$

Finally, we solve for u_v to get the final control law

$$u_{v,ct} = \begin{cases}\frac{\alpha_4\tau_v V}{\rho_0 C_v ART_0\psi(\frac{p}{P_S})P_S}, & q_v > 0\\[2ex] \frac{\alpha_4\tau_v V}{\rho_0 C_v ART_0\psi(\frac{P_E}{p})p}, & q_v \leq 0\end{cases} \tag{5.2}$$

which is also the basis of the discrete-time controllers presented in the next section.

5.3.1 Stability Properties

The Lyapunov function V_4 is positive definite and radially unbounded $\forall e_i \neq 0,\ i = 1, 2, 3, 4$. The derivative $\dot{V}_4$ is negative definite $\forall e_i \neq 0,\ i = 1, 2, 3, 4$. Hence, V is non-increasing and converges as $t \to \infty$. V is bounded below by zero, and the error variables e_i are bounded. Since the reference trajectory y_r and its derivatives are bounded by assumption, the system states are bounded. It can be shown, using the Comparison Lemma (see Appendix A) that the error dynamics converge to zero exponentially, thus $\lim_{t\to\infty} y(t) = y_r(t)$. Hence, the controller achieves:

- exponential tracking for the position y in the region of validity Ω_v.
- all system states remain bounded.

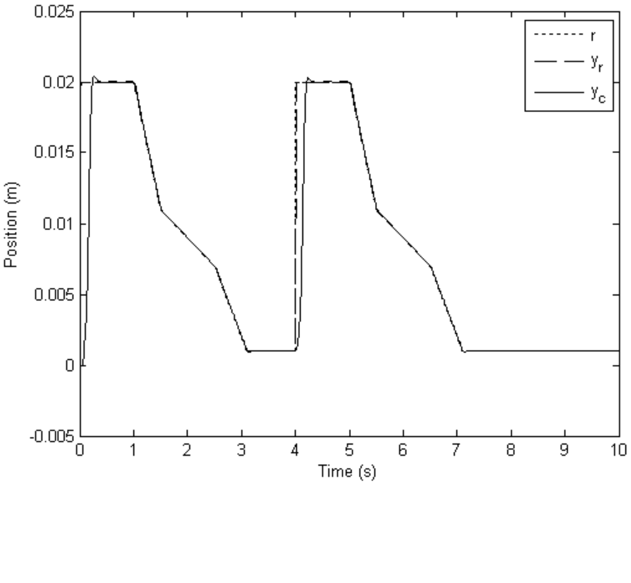
The design parameters available for the controller are $c_i > 0$ for $i = 1, 2, 3, 4$.

5.3.2 Simulation

The continuous-time controller for the fifth order model is simulated to prove good tracking performance of the clutch sequence, as shown in figure 5.1. We see that the reference filter renders a smooth trajectory for tracking control, but due to the introduction of the flow and valve dynamics, the system reacts somewhat slower than the third order system in the previous chapter.

The design constants for the continuous-time backstepping controller is chosen as $c_1 = c_2 = 800,\ c_3 = 30,\ c_4 = 350$. Looking at the system states and input in figure 5.2 we see that they are all well within reasonable physical limits, causing the system to respond slower than in an unbounded case.

For a controller to be realizable it is crucial that it exploit its system knowledge to produce smooth actuation and not introduce chattering in the control input, the latter minimizing the probability for the controller to work in practice and lowering its chances for rendering good performance. The control input used by the controller clearly utilizes the system information to produce smooth input through the critical region, but the small oscillations we observe in the other regions of the reference trajectory give unecessary wear and tear, and could potentially reduce the lifetime of the clutch.



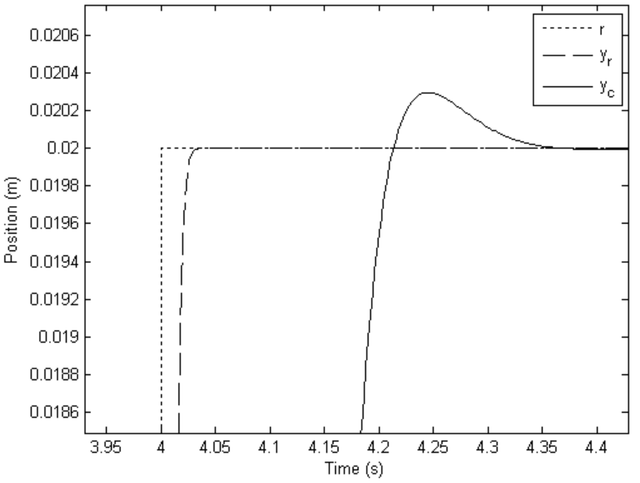


Figure 5.1: Simulation of the continuous-time controller

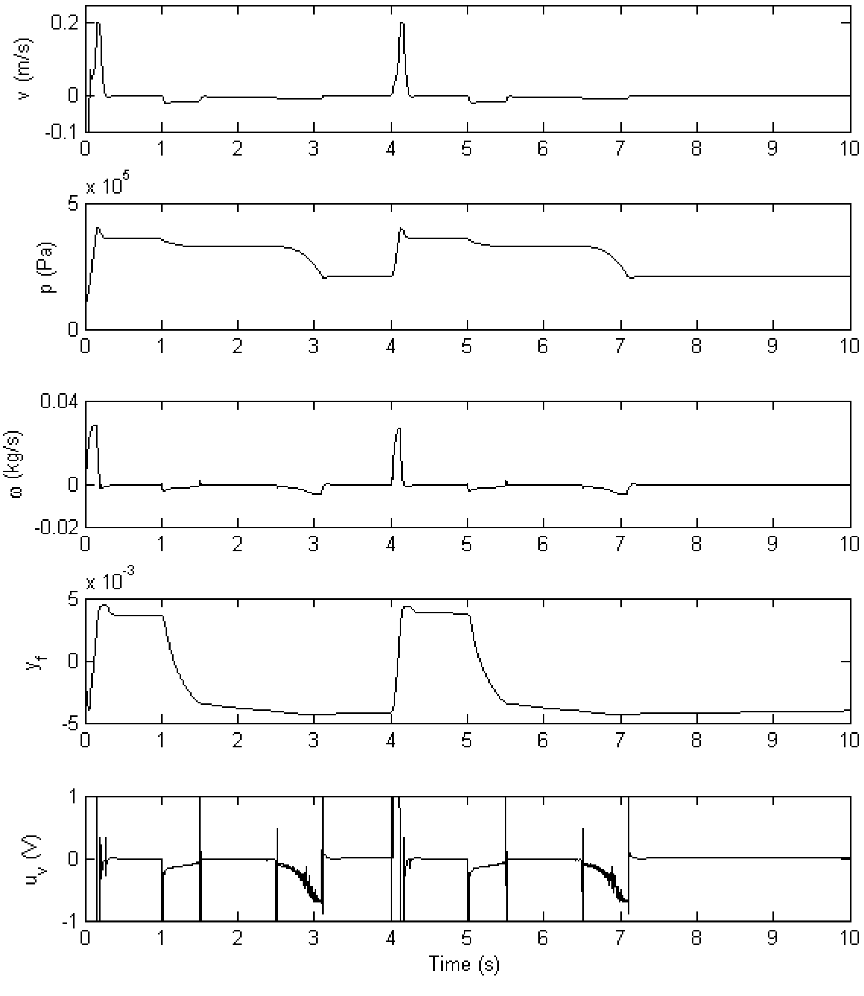


Figure 5.2: System states and input

5.4 Discrete-Time Backstepping

This section follows the procedure of Section 3.4 to present the two methods for backstepping in the discrete-time domain, starting with the simple emulation method.

5.4.1 Emulation Design

For the fifth order model, the emulation controller is

$$u_{v,em} = u_{v,ct}(k), \tag{5.3}$$

that achieves global asymptotic stability of the closed loop system (5.1), (5.3) according to the general and unified framework presented in Laila (2003). Next, the controller obtained from the direct discrete-time design is presented.

5.4.2 Direct Discrete-Time Design

The fifth order Euler approximate model of the system is

$$\begin{aligned}
y(k+1) &= y(k) + Tv(k) \\
v(k+1) &= v(k) + \frac{T}{M}(-f_l(y(k)) - f_f(v(k)) - AP_0 + Ap(k)) \\
\zeta(k+1) &= \zeta(k) + TRT_0\omega(k) \\
\omega(k+1) &= \omega(k) + \frac{T}{\tau_v}\Big[-\omega(k) + q_v(p(y(k),\zeta(k)),u_v(k))\Big] \\
y_f(k+1) &= y_f(k) + Tg_f(v(k),y_f(k))
\end{aligned} \tag{5.4}$$

For the sake of simplicity the arument (k) is omitted from the rest of this section, just remember that we operate in the discrete-time domain.

The parameters for the discrete-time backstepping procedure is obtained from the third step of the continuous-time backstepping procedure

$$\begin{aligned}
\alpha_T &= K_5 y + K_6 v + \frac{\partial f_l(y)}{\partial y}v + c_3 f_l(y) + c_3 f_f(v,y_f) - c_3 Ap + c_3 AP_0 + \frac{A^2}{V^2}\zeta v + \frac{\partial f_f(v,y_f)}{\partial y_f}g_f(v,y_f) \\
&\quad + \frac{1}{M}\left[K_2 + \frac{\partial f_f(v,y_f)}{\partial v}\right](-f_l(y) - f_f(v,y_f) + Ap - AP_0) + K_7 y_r + K_8\dot{y}_r + K_9\ddot{y}_r + My_r^{(3)} \\
W_T &= \frac{1}{2}\Big[(y-y_r)^2 + M(v + c_1(y) - \dot{y}_r + c_1 y_r)^2 \\
&\quad + (Ap - M\ddot{y}_r - K_1 y - K_2 v - K_3 y_r - K_4\dot{y}_r - f_l(y) - f_f(v,y_f) - AP_0)^2\Big],
\end{aligned}$$

Then, following the steps described in Section 3.3 (for simplicity, we write $\dot{y}, \dot{v}, \dot{\zeta}$ and $\dot{y}_f$ instead of the complete equations)

$$
\begin{aligned}
\Delta\alpha_T = T\Big[& K_5\dot{y} + K_6\dot{v} + c_3\frac{\partial f_l(y)}{\partial y}\dot{y} + \frac{\partial^2 f_l(y)}{\partial y^2}v\dot{y} + \frac{\partial f_l(y)}{\partial y}\dot{v} + c_3\frac{\partial f_l(y)}{\partial y}\dot{y} + c_3\frac{\partial f_f(v,y_f)}{\partial v}\dot{v} \\
& + c_3\frac{\partial f_f(v,y_f)}{\partial y_f}\dot{y}_f - c_3\frac{A}{V}\dot{\zeta} + c_3\frac{A^2}{V^2}\zeta\dot{y} + \frac{A^2}{V^2}\zeta\dot{v} + \frac{A^2}{V^2}v\dot{\zeta} - \frac{2A^3}{V^3}\zeta v\dot{y} \\
& + \frac{\partial^2 f_f(v,y_f)}{\partial v\partial y_f}g_f(v,y_f)\dot{v} + \frac{\partial^2 f_f(v,y_f)}{\partial y_f^2}g_f(v,y_f)\dot{y}_f + \frac{\partial f_f(v,y_f)}{\partial y_f}\frac{\partial g_f(v,y_f)}{\partial v}\dot{v} \\
& + \frac{\partial f_f(v,y_f)}{\partial y_f}\frac{\partial g_f(v,y_f)}{\partial y_f}\dot{y}_f + \frac{\partial^2 f_f(v,y_f)}{\partial v^2}(-f_l(y) - f_f(v,y_f) + Ap - AP_0)\dot{v} \\
& + \frac{\partial^2 f_f(v,y_f)}{\partial y_f\partial v}(-f_l(y) - f_f(v,y_f) + Ap - AP_0)\dot{y}_f + \frac{1}{M}\left[K_2 + \frac{\partial f_f(v,y_f)}{\partial v}\right]\left(-\frac{\partial f_l(y)}{\partial y} - \frac{A^2}{V^2}\zeta\right)\dot{y} \\
& - \frac{1}{M}\left[K_2 + \frac{\partial f_f(v,y_f)}{\partial v}\right]\frac{\partial f_f(v,y_f)}{\partial v}\dot{v} - \frac{1}{M}\left[K_2 + \frac{\partial f_f(v,y_f)}{\partial v}\right]\frac{\partial f_f(v,y_f)}{\partial y_f}\dot{y}_f \\
& + \frac{1}{M}\left[K_2 + \frac{\partial f_f(v,y_f)}{\partial v}\right]\frac{A}{V}\dot{\zeta} + K_7\dot{y}_r + K_8\ddot{y}_r + K_9 y_r^{(3)} + My_r^{(4)}\Big] \\
\Delta\alpha_T = T(& \dot{\alpha}_3 + My_r^{(4)})
\end{aligned}
$$

and

$$
\begin{aligned}
\overline{\Delta W}_T = {} & \frac{1}{2}\Big[Ap + T\left(A\frac{RT_0}{V}\omega - \frac{A^2}{V^2}\zeta v\right) - M(\ddot{y}_r + Ty_r^{(3)}) - K_1(y + Tv) - K_2(v + T\dot{v}) - K_3(y_r + T\dot{y}_r) \\
& - K_4(\dot{y}_r + T\ddot{y}_r) - f_l(y + Tv) - f_f(v + T\dot{v}, y_f) - f_f(v, y_f + T\dot{y}_f) - AP_0\Big]^2 \\
& - \frac{1}{2}\Big[Ap + T\left(\alpha_T - \frac{A^2}{V^2}\zeta v\right) - M(\ddot{y}_r + Ty_r^{(3)}) - K_1(y + Tv) - K_2(v + T\dot{v}) - K_3(y_r + T\dot{y}_r) \\
& - K_4(\dot{y}_r + T\ddot{y}_r) - f_l(y + Tv) - f_f(v + T\dot{v}, y_f) - f_f(v, y_f + T\dot{y}_f) - AP_0\Big]^2 \\
& = \Big[e_3 + \frac{T}{2}\Big(\frac{ART_0}{V}\omega + K_5 y + K_6' v - \frac{\partial f_l(y)}{\partial y}v + K_7 y_r + K_8'\dot{y}_r + K_9'\ddot{y}_r - My_r^{(3)} - \frac{A^2}{V^2}\zeta v - c_3 Ap \\
& + c_3 f_l(y) + c_3 f_f(v,y_f) + c_3 AP_0 - \frac{\partial f_f(v,y_f)}{\partial y_f}g_f(v,y_f) - \left[K_2 + \frac{\partial f_f(v,y_f)}{\partial v}\right]\dot{v}\Big)\Big] \\
& \cdot T\left[\frac{ART_0}{V}\omega - \alpha_T\right] \qquad \Rightarrow \qquad \widetilde{\Delta W}_T = T\left(e_3 + \frac{T}{2}DISC\right)
\end{aligned}
$$

where

$$
K_6' = -1 + c_3K_2 - K_1, \quad K_8' = 1 + c_3K_4 - K_3, \quad K_9' = c_3M - K_4
$$

and

$$
\begin{aligned}
DISC = {} & \frac{ART_0}{V}\omega + K_5 y + K_6' v - \frac{\partial f_l(y)}{\partial y}v + K_7 y_r + K_8'\dot{y}_r + K_9'\ddot{y}_r - My_r^{(3)} - \frac{A^2}{V^2}\zeta v - c_3 Ap \\
& + c_3 f_l(y) + c_3 f_f(v,y_f) + c_3 AP_0 - \frac{\partial f_f(v,y_f)}{\partial y_f}g_f(v,y_f) - \left[K_2 + \frac{\partial f_f(v,y_f)}{\partial v}\right]\dot{v}.
\end{aligned}
$$

This provides the control law

$$
u = -c_4\left(\frac{ART_0}{V}\omega - \alpha_T\right) - \frac{\widetilde{\Delta W}_T}{T} + \frac{\Delta\alpha_T}{T} = -c_4 e_4 - e_3 - \frac{T}{2}DISC + My_r^{(4)} + \dot{\alpha}_3.
$$

Then, by definition

$$\begin{aligned}\left(\frac{ART_0}{V}\omega\right)' &= \frac{ART_0}{\tau_v V}(-\omega + q_v) - \frac{A^2RT_0}{V^2}\omega v = u \quad \Rightarrow \\ \frac{ART_0}{\tau_v V}q_v &= u + \frac{ART_0}{V}\left(\frac{A}{V}v + \frac{1}{\tau_v}\right)\omega = \alpha_4 - \frac{T}{2}DISC\end{aligned}$$

which leads us to the final control law

$$u_{v,dt} = \begin{cases} \frac{\tau_v V}{\rho_0 C_v ART_0 \psi(\frac{p}{P_S})P_S}\left(\alpha_4 - \frac{T}{2}DISC\right), & q_v > 0 \\ \\ \frac{\tau_v V}{\rho_0 C_v ART_0 \psi(\frac{P_E}{p})p}\left(\alpha_4 - \frac{T}{2}DISC\right), & q_v \leq 0 \end{cases} \tag{5.5}$$

which achieves semiglobal-practical stability for the approximate discrete-time model (5.4), (5.5), and furthermore asymptotic stability of the closed loop sampled data-system (5.1), (5.5) in the region of validity Ω_v. This can be proven by the Lyapunov function $V_T = W_T + \frac{1}{2}(\frac{ART_0}{V}\omega - \alpha_T)^2$.

5.5 Controller Comparison

This section presents simulation results of the three controllers (5.2), (5.3) and (5.5). The controllers are tested for different sampling intervals, to investigate their trajectory tracking performance under different circumstances. The output from the continuous-time system y_c is used as a reference for the position output of the sampled-data systems, as in the last chapter.

The aim of this section is to compare the results from the two sampled-data systems in order to conclude on the performance of the discrete-time controllers, and we hope to see some difference when using slower sampling. Although, remembering the results from last chapter where it seemed that the complex dynamics of the clutch actuator cause the need for fast sampling, in which the two controllers are expected to perform equally.

5.5.1 Simulation with T = 2ms

This controller comparison utilizes some of the same sampling intervals as in Section 4.5, starting with $T = 2ms$ as shown in figure 5.3. Both the discrete-time controllers perform well for this low sampling interval, as expected. More interesting is the fact that, as opposed to for the third order model, the two controllers have somewhat different trajectories as shown in the zoomed-in plot.

All three controllers use the same set of design constants for this sampling interval; the set stated in Subsection 5.3.2. However, how the controllers perform for this sampling interval is somewhat trivial, and we quickly move on to higher sampling intervals.

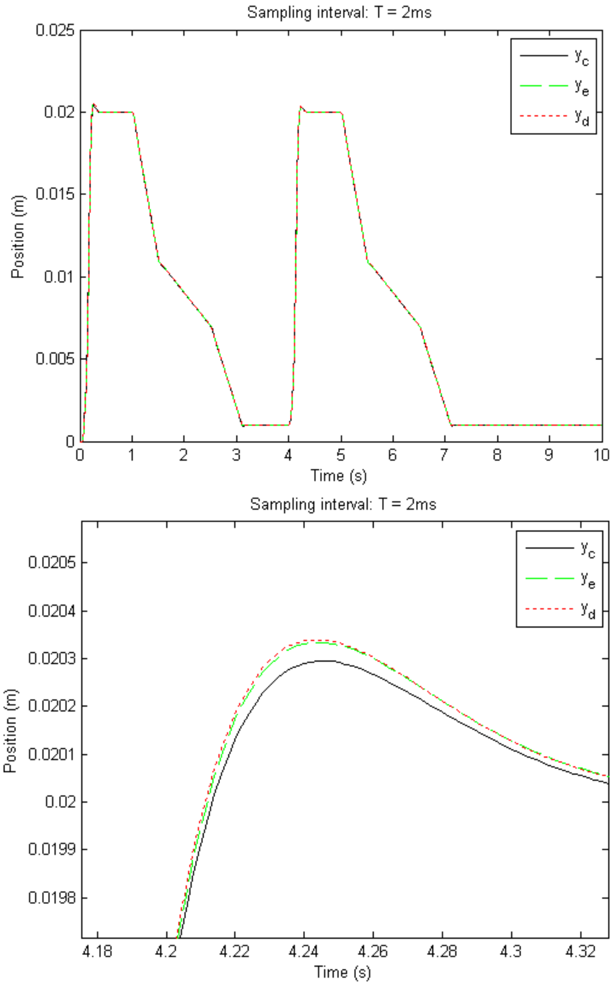


Figure 5.3: Controller comparison, T = 2ms

5.5.2 Simulation with T = 10ms

When using higher sampling intervals for the fifth order sampled-data systems, small oscillations occur in the response of the system, as shown in figure 5.4. The two controllers cause somewhat different trajectories as we can se from the plot, but clearly it is impossible to state that one is better than the other. The oscillations proved to be difficult to remove, and it seems to occur due to the introduction of the valve dynamics.

The oscillations seem to happen after the reference trajectory changes rapidly, e.g. when the clutch is disengaged. Looking at the system states and input for the sampled-data system with the direct discrete-time designed controller in figure 5.5, we see that also the other states have small oscillations in their response. Clearly, this is caused by the control input voltage, that acts in a highly unrealistic way with its switching from $-1V$ to $1V$ all the time. Since we would never get a response like that with the physical system, the controllers should be implemented in the test rig to observe differences as the sampling interval is chosen higher.

Since we do not get any reasonable results from the comparisons with higher sampling intervals, we will instead look at the contributions from the continuous- and discrete-time terms in the Euler based controller, as we did in the last chapter.

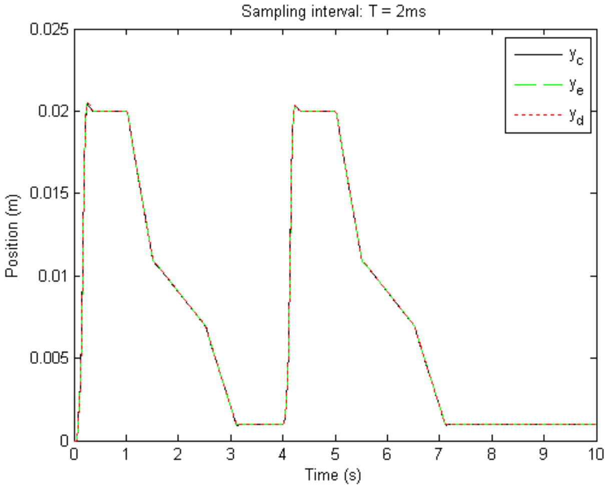


Figure 5.4: Controller comparison, T = 10ms

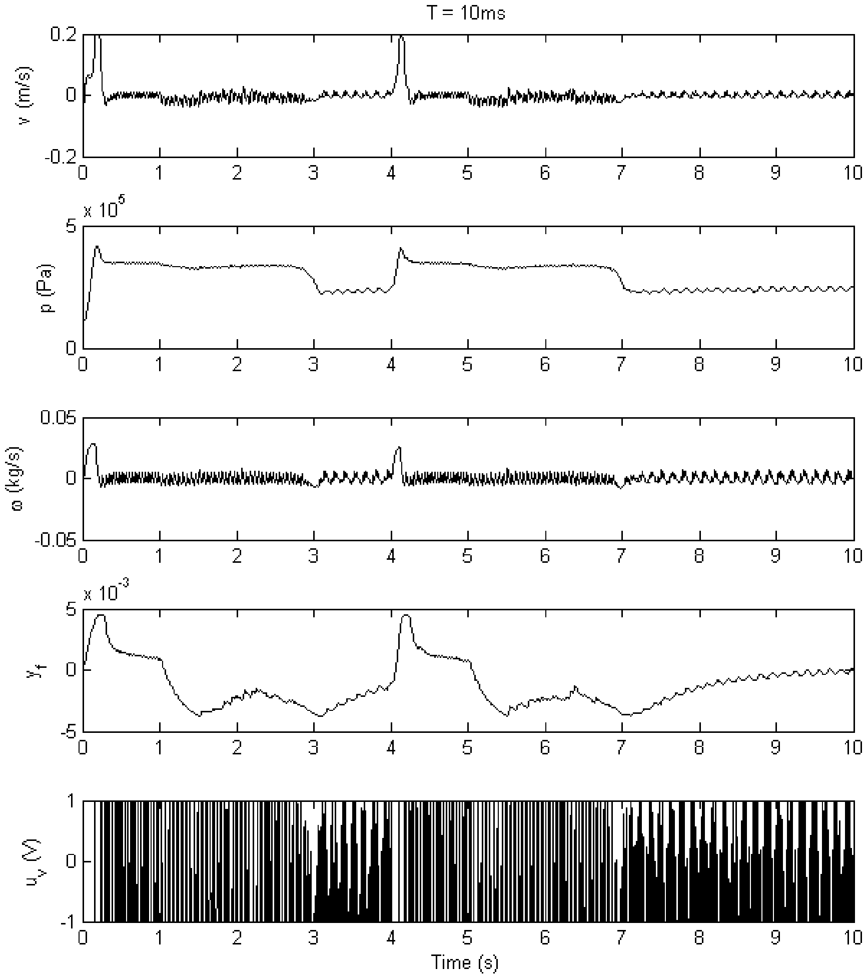
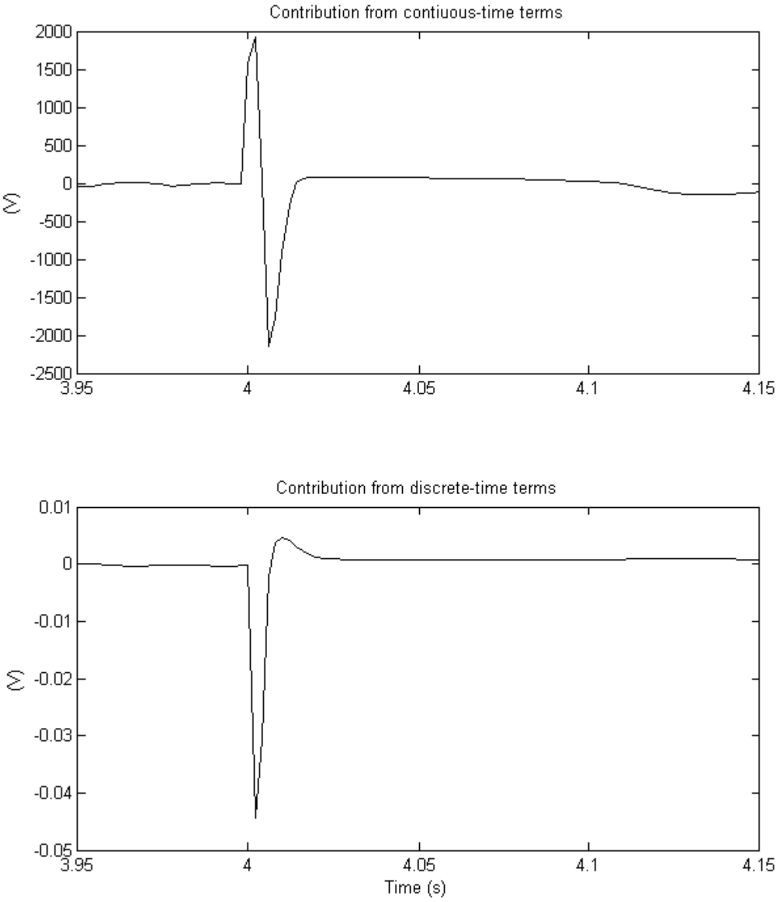


Figure 5.5: System states and input, sampled-data model

5.6 Conclusion

Since the controller comparison itself did not provide a clear answer, we look at the contributions from the two designs in figure 5.6. We see the terms from the continuous-time design totally dominate the terms from the direct discrete-time design, thus making that the latter negligible for the sampling intervals the clutch actuator operates within, and causing the two discrete-time controllers to perform in a similar manner.

It is likely that it is the considerable forces of the load and friction, and especially the introduction of their derivatives into the continuous-time part of the controller that makes the huge difference in value of the continuous-time and discrete-time terms. This leads us to believe that whatever properties of adjustment the Euler-based controller might hold is lost in the complexity of the design, at least for the fast sampling required for our system. Some sort of improvements might occur for really high sampling intervals as the discrete-time terms grow larger, but those sampling intervals would certainly be useless in practice.


Contribution from contiuous-time terms

Contribution from discrete-time terms

Figure 5.6: Contributions from continuous- and discrete-time terms in controller u_{dt}

The ultimate test, however, would be to implement the controllers on the physical system to observe their performance when tested under realistic conditions. In either way, it should always be solely positive to account for the sampling of the system when designing one's controller. We have seen from examples that this design has the potential to improve stability and performance, and worst case scenario is just that no difference in controller performance is obtained.

Chapter 6

Discussion and Further Work

As we have seen, the analysis of sampled-data systems leads to new challenging control problems with a rich mathematical structure. Many of these problems have been solved only very recently, and some of the most important results are presented in this thesis. The literature study conducted have shown that the ultimate method of controlling sampled-data systems would be to use the exact sampled-data model of the plant and design a controller that achieves stability and required performance for the closed loop system sampled-data system. However, since results on this method are scarce and in general hard to obtain, it appears that the direct discrete-time design provides the best alternative available.

The direct discrete-time design is done by designing a controller for the Euler approximate discrete-time model of the plant, and by using results which prove that for given conditions the stabilization of the approximate discrete-time system also ensures stabilization of the exact discrete-time system and thereafter also the sampled-data system. This design technique has proved potential to improve performance of nonlinear sampled-data systems compared to using an ad hoc discretization of a continuous-time controller.

As for the electro-pneumatic clutch actuator, presented here with both third and fifth order models, the digital implementation of the continuous-time controller clearly decrease the performance of the system, making it respond somewhat slower with a little overshoot and some oscillations in its response. But, however promising the direct discrete-time design was made out to be, we have not been able to spot any difference in performance from implementing it instead of the simple emulation design. This could be because of the relative complexity of the design model, and the fact that the introduction of the load and friction forces and their derivatives makes the continuous-time parts of the controller hugely dominating the discrete-time parts, thus the later being negligible for the fast sampling the clutch actuator operates with. For slower sampling, however, the discrete-time terms get bigger and might make a difference if the sampling interval was to be chosen as some hundred milliseconds. This is only of academic interest though, since these sampling intervals could never be used in practice for the clutch actuator system.

On the topic of further work, it would be interesting to see how the controllers perform on the physical test-system, where unmodelled dynamics and measurement noise could cause the system to respond in uncertain ways. Whether we would experience any improvement on the trajectory tracking properties or not is in reality a trivial question, since any controller implemented on a sampled-data system *should* be designed with the sampling of the system in mind.

Recommended for further work is also using a cascade connection of an observer and the controller for output feedback control instead of full state feedback control. An observer acts as a low-pass filter on the state measurements and may help filter out the oscillations we experienced at the fifth order model.

There are some other methods for designing a controller that accounts for the sampling of the system that could be interesting to pursue. The Lyapunov redesign method has already been mentioned, and there are a couple of other results on continuous-time nonlinear controller redesign for sampled-data implementations, like Nesic et al. (2005) that present a framework for redesigning continuous-time controllers based on Fliess series expansions.

Furthermore, Nesic et al. (2006) also presents, in addition to the SPA stabilizing pair described in Section 3.3, a GA stabilizing pair for the Euler model that is supposedly harder to compute, but might still be worth to look at. Since the topic of control of nonlinear sampled-data system is still in its earliest phase of research, it is quite possible that there will be presented alternative results in the next few years, which should give more possibilites for controller design.

Bibliography

Chen, T. & Francis, B. A. (1995), *Optimal Sampled-Data Control Systems*, Springer-Verlag.

Kaasa., G. O. (2003), *Nonlinear output-feedback control applied to electro-pneumatic clutch actuation in heavy-duty trucks*, PhD thesis, NTNU.

Khalil., H. (1996), *Nonlinear Systems*, Prentice Hall.

Krstić, M., Kanellakopoulos, I. & Kokotovic, P. (1995), *Nonlinear and Adaptive Control Design*, Wiley Interscience.

Laila., D. (2003), *Design and analysis of nonlinear sampled-data systems*, PhD thesis, The University of Melbourne.

Lokken, K. (2006), 'Adaptive backstepping with application with electro-pneumatic clutch actuators'.

Nesic, D., Astolfi, A. & Laila, D. (2005), 'Advanced topics in control systems theory ii: Lecture notes from fap 2005'.

Nesic, D. & Grune, L. (2005), 'Lyapunov based continous-time controller redesign for sampled-data implementation'.

Nesic, D., Kokotovic, P. & Teel, A. (2000), 'Sufficient conditions for stabilization of samped-data nonlinear systems via discrete-time approximations'.

Nesic, D. & Teel, A. (2001), 'Backstepping on the euler approximate model for stabilization of sampled-data nonlinear systems'.

Nesic, D. & Teel, A. (2004), 'A framework for stabilization of samped-data nonlinear systems based on their approximate discrete-time models'.

Nesic, D. & Teel, A. (2006), 'Stabilization of sampled-data nonlinear systems via backstepping on their euler approximate model'.

Appendix A

Definitions and Theorems

In this appendix some useful theorems, lemmas and definitions that are referred to in this thesis are stated. All of the mathematic results below are taken from various references in the Bibliography.

A.1 Mathematical Preliminaries

Definition 3. *A function $\gamma : \mathcal{R}_{\geq 0} \rightarrow \mathcal{R}_{\geq 0}$ is of class-$\mathcal{K}$($\gamma \in \mathcal{K}$) if it is continous strictly increasing and $\gamma(0) = 0$. It is of class-$\mathcal{K}_\infty$ if it in addition is unbounded. Funcitons of class-$\mathcal{K}_\infty$ are invertible. A function $\beta : \mathcal{R}_{\geq 0} \rightarrow \mathcal{R}_{\geq 0}$ is of class-$\mathcal{KL}$ if $\beta(\cdot, t)$ is of class-$\mathcal{K}$ for each $t \geq 0$ and $\beta(s, \cdot)$ is decreasing to zero for each $s > 0$.*

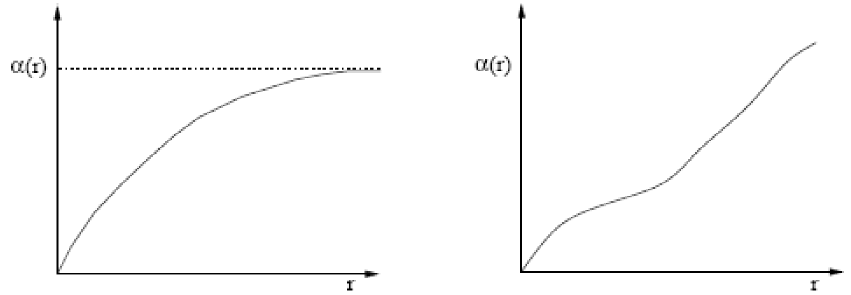


Figure A.1: Class $\mathcal{K}$ and class $\mathcal{K}_\infty$, respectively

Theorem 4 (Mean Value Theorem). *Assume that $f : \mathcal{R}^n \rightarrow \mathcal{R}$ is continously differentiable at each point x of an open set $S \subset \mathcal{R}^n$. Let x and y be two points of S such that the line segment $L(x, y) \subset S$. Then, there exists a point z of $L(x, y)$ such that*

$$f(y) - f(x) = \frac{\partial f}{\partial x}|_{x=z}(y - x).$$

The line segment $L(x, y)$ joining two distinct points x and y in $\mathcal{R}^n$ is

$$L(x, y) = \{z | z = \theta x + (1 - \theta)y, \;\; 0 < \theta < 1\}$$

■

Lemma 2 (Comparison Lemma). *Consider the scalar differential equation*

$$\dot{u} = f(t, u), \;\; u(t_0) = u_0$$

where $f(t, u)$ is continuous in t and locally Lipschitz in u for all $t \geq 0$ and all $u \in \mathcal{J} \subset \mathcal{R}$. Let $[t_0, T)$ be the maximal interval of existence of the solution $u(t)$, and suppose $u(t) \in \mathcal{J}$ for all $t \in [t_0, T)$. Let $v(t)$ be a continuous function whose upper right-hand derivative D^+ satisfies the differential inequality

$$D^+ v(t) \leq f(t, v(t)), \;\; v(t_0) \leq u_0$$

with $v(t) \in \mathcal{J}$ for all $t \in [t_0, T)$. Then $v(t) \leq u(t)$ for all $t \in [t_0, T)$. ■

A.2 SPA Stability

Definition 4 (SPA stability). *The family of systems $x(k+1) = F_T(x(k), u_T(x(k)))$ is SPA stable if there exists $\beta \in \mathcal{KL}$ such that for any strictly positive real numbers (Δ, δ) there exists $T^\star > 0$ such that for all $T \in (0, T^\star)$, all initial states $x(0) = x_0$ with $|x_0| \leq \Delta$, the solutions of the system satisfy*

$$|x(k)| \leq \beta(|x_0|, kT) + \delta, \;\; \forall k \in \mathcal{N}$$

■

Definition 5 (SPAS Lyapunov function). *A continously differentiable function $V_T : \mathcal{R}^n \rightarrow \mathcal{R}$ is called SPAS Lyapunov function for the system F_T if there exists class $\mathcal{K}_\infty$ functions α, β, γ such that for any strictly positive real numbers (Δ_x, ν), there exists $L, T^\star > 0$ such that for all $T \in (0, T^\star)$ and for all $x, y \leq \Delta_x$ the following holds.*

$$\begin{aligned} \alpha(|x|) \leq V_T(x) &\leq \beta(|x|) \\ V_T(F_T(x, u_T(x))) - V_T(x) &\leq -T\gamma(|x|) + T\nu \\ |V_T(x) - V_T(y)| &\leq L|x - y| \end{aligned}$$

In this case, we say that the pair (V_T, u_T) is Lyapunov SPA stabilizing for the system F_T. ■

Definition 6 (One-step consistency). *The family F_T^a is said to be one-step consistent with F_T^e if there exists functions $\rho, \varphi_1, \varphi_2 \in \mathcal{K}_\infty$ such that given any strictly positive real numbers (Δ_x, Δ_u) there exists $T^\star > 0$ such that, for all $T \in (0, T^\star), |x| \leq \Delta_x, |u| \leq \Delta_u$ we have*

$$|F_T^e(x, u) - F_T^a(x, u)| \leq T\rho(T)[\varphi_1(|x|) + \varphi_2(|u|)].$$

■

Definition 7. *The family of controllers u_T is bounded, uniformly in small T, if there exist $\kappa \in \mathcal{K}_\infty$ and for any $\Delta > 0$ there exists $T^\star > 0$ such that for all $|x| \leq \Delta$ and $T \in (0, T^\star)$ we have*

$$|u_T(x)| \leq \kappa(|x|).$$

■

Using the above definitions we can now state the following result.

Theorem 5. *Suppose the following conditions hold.*

1. F_T^a *is one-step consistent with* F_T^e.
2. u_T *is bounded, uniformly in small* T.
3. *There exists a SPAS Lyapunov function for the system* (F_T^a, u_T).

Then, the system (F_T^e, u_T) *is SPA stable, and hence, the sampled-data system* $\dot{x}(t) = f(x(t), u_T(x(kT))))$, $t \in [kT, (k+1)T]$ *is SPA stable.* ■

A.3 ISS Stability

Input-to-state stability is a very important property of nonlinear systems with inputs.

Definition 8 (Input-to-state (ISS) stability). *The nonlinear system* $\dot{x} = f(x, u)$ *is said to be input-to-state stable if there exist a class* $\mathcal{KL}$ *function* $\beta(\cdot,\cdot)$ *and a class* $\mathcal{K}$ *function* $\gamma(\cdot)$, *called a gain function, such that for any input* $u \in \mathcal{K}_\infty$ *and any* $x_0 \in \mathcal{R}^n$, *the response* $x(t)$ *of the system in the initial state* $x(0) = x_0$ *satisfies*

$$|x(t)| \leq \beta(|x_0|, t) + \gamma(||u||_\infty)$$

for all $t \geq 0$. ■

Theorem 6. *A continous and differentiable function* $V : \mathcal{R}^n \to \mathcal{R}$ *is called an ISS Lyapunov function for the system* $\dot{x} = f(x, u)$ *if there exists class* $\mathcal{K}_\infty$ *functions* α, β, γ *and a class* $\mathcal{K}$ *function* χ *such that*

$$\alpha(|x|) \leq V(x) \leq \beta(|x|)$$
$$|x| \geq \chi(|u|) \Rightarrow \frac{\partial V}{\partial x} f(x, u) \leq -\gamma(|x|)$$

for all $x \in \mathcal{R}^n$ *and all* $u \in \mathcal{R}^m$ ■

Theorem 7. $\dot{x} = f(x, u)$ *is input-to-state stable if and only if there exists an ISS Lyapunov function for the system.* ■

Appendix B

Partial Derivatives of the Lugre Friction Model

The partial derivatives of the Lugre friction model that emerges in the backstepping procedures of Chapter 5 is stated here for reference, and in case of further work.

$$\begin{aligned}
\frac{\partial f_f(v,y_f)}{\partial v} &= D_v + D_f - \frac{D_f K_f}{F_d} y_f \frac{v}{|v|_s} \\
\frac{\partial f_f(v,y_f)}{\partial y_f} &= K_f - \frac{D_f K_f}{F_d} |v|_s \\
\frac{\partial^2 f_f(v,y_f)}{\partial v^2} &= -\frac{D_f K_f}{F_d} y_f \frac{\epsilon^2}{|v|_s \cdot |v|_s^2} \\
\frac{\partial^2 f_f(v,y_f)}{\partial y_f^2} &= 0 \\
\frac{\partial^2 f_f(v,y_f)}{\partial v \partial y_f} &= \frac{\partial^2 f_f(v,y_f)}{\partial y_f \partial v} = -\frac{D_f K_f}{F_d} \frac{v}{|v|_s}
\end{aligned}$$

where $|v|_s = \sqrt{v^2 + \epsilon^2}$ as explained in Chapter 5.