Norwegian University of
Science and Technology

Control Design for Backstepping and Adaptive Backstepping Controllers in MPD

Martin Bergene Johansen

Master of Science in Cybernetics and Robotics
Submission date: June 2017
Supervisor: Tor Arne Johansen, ITK
Co-supervisor: Glenn-Ole Kaasa, Kelda Drilling Controls
Jon Åge Stakvik, Kelda Drilling Controls

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Preface

This is a master thesis as part of the Cybernetics and Robotics study program at NTNU during the spring of 2017. This project has been carried out in cooperation with Kelda Drilling Controls.

I have had the pleasure of working for Kelda the past summers and last semester I wrote a preparation project on "dynamic modeling for pump flow estimations in MPD operations" in cooperation with Kelda. It was decided with Jon Åge Stakvik from Kelda to change the intended master in order him to able to use the results from my master thesis in his PhD study. Kelda has an ambition to be a leading company in the drilling industry and to achieve this they focus greatly on research. It is intended that the results from this thesis will help both Kelda and Jon Åge Stakvik in their research.

This is a thesis about the development of controllers used in Managed Pressure Drilling operations in the field of oil drilling. It is expected that the reader is familiar with this field and has a basic knowledge of Managed Pressure Drilling operations, but a short introduction of Managed Pressure Drilling is given in the introduction.

Trondheim, 2017-06-12

(Your signature)

Martin Bergene Johansen

Acknowledgment

I would like to thank Professor Tor Arne Johansen for being my main supervisor in this project.

Thanks to Professor Kristin y. Pettersen, NTNU, and senior researcher Antonio Loria, CNRS, for their guidance in stabilization of systems using cascade theory, and to Professor Ole Morten Aamo, NTNU, for his help on adaptive backstepping.

I would also like to thank Glenn-Ole Kaasa, CEO at Kelda Drilling Controls, for making this project possible and for for the cooperation the past 3 years.

Last, but not least, I would like to thank Jon Åge Stakvik, research engineer at Kelda Drilling Controls, for his excellent guidance during this project. His availability when I had problems that needed discussion, his expertise in Managed Pressure Drilling and control theory, and his positive attitude has proven more than helpful in this project.

M.B.J.

NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for informasjonsteknologi, matematikk og elektroteknikk
Institutt for teknisk kybernetikk

Master of Science Thesis Assignment

Name of the candidate: Martin Bergene Johansen
Subject: Engineering Cybernetics
Title: Control design for backstepping and adaptive backstepping controllers in MPD

Background

Managed Pressure Drilling (MPD) is an automatic control technique used to control the pressure in an oil well. Current state-of-the art solutions in the industry are, in general, built on PI control design of a linearized model that requires accurate tuning and might be restricted to operate in a neighborhood of the linearized parameters. In all MPD setups, the well pressure is controlled by one or more choke valves. The actuators driving the choke valve have an internal speed controller that requires the MPD system to provide an actuator speed reference to control the choke valve position. This creates a model with a recursive structure that has potential to be stabilized and controlled with a backstepping control design approach. In this work, the candidate will investigate a backstepping design for MPD control with initial focus on choke pressure control and possible extension to bottom hole pressure control.

Assignment:

1. Perform a literature review on backstepping control design, with focus on adaptive backstepping and tracking control.
2. Develop a design model suited for backstepping design in MPD. The control input to be considered is the actuator speed. Verify the design model on field and simulator data provided by Kelda Drilling Controls. Perform an analysis of uncertain parameters and error terms in the design model.
3. Design a backstepping controller for MPD where the actuator speed is the control input.
4. Design an adaptive backstepping controller for MPD that estimates commonly unknown parameters. The actuator speed is the control input.
5. Simulate and analyze the performance of the backstepping and adaptive backstepping controller in relevant MPD scenarios performed on a high fidelity simulator. Verify performance in case of model mismatch between the design model and simulator.

To be handed in by: 7/7-2017
Co-supervisors: Dr. Glenn-Ole Kaasa, PhD-student Jon Åge Stakvik

Trondheim, 06.01.2017

____

Tor Arne Johansen
Professor, supervisor

Summary and Conclusions

In the field of oil drilling there is a concept known as Managed Pressure Drilling. In Managed Pressure Drilling one relies on pumps in combination with chokes (valves) in order to maintain a certain pressure at the bottom of the well. It is crucial to maintain this pressure since too low pressure may cause the well to collapse and too high pressure may cause the well to fracture, making it more susceptible for a collapse. The pressure at the bottom of the well is mainly controlled by a choke.

In this thesis controllers based on backstepping theory will be developed in order to control the choke pressure. The controllers will be designed such that the choke pressure will track a reference value and the controllers will achieve this by adjusting the choke's angular velocity. It is also common that certain parameters in the well are unknown due to changes that occur in the fluid running through the choke. Because of this, adaptive integrator backstepping controllers will also be developed. The goal for this project is to obtain an understanding of how well controllers based on integrator backstepping and adaptive integrator backstepping can solve this task.

Integrator backstepping controllers are useful when designing controllers for systems on cascade form of order 2 or higher, however, the complexity of these controllers increases significantly as the order of the system increases. Because of this, two different design models have been used in this thesis in order to develop controllers based on integrator backstepping. One is of 2nd order and the other is of 3rd order. The 2nd order design model describes how the pressure changes according to the flow in and out of the choke. The flow out of the choke depends on the choke opening and the choke opening changes based on the angular velocity of the choke. The 3rd order design model is an extension of the 2nd order design model. The 3rd order model also describes the actuator dynamics in the choke which will provide a better representation on how the angular velocity affects the choke position.

A controller was first developed for the 2nd order system. If the controller behaved well, a new controller would be developed for the 3rd order system. In order to verify the performance of the controllers, they were simulated in common scenarios that occur during drilling. Controllers that performed well during these simulations went through further testing by perform-

ing simulations with Straume®, a high-end multiphase well simulator. Straume® provides a better representation of a real life drilling scenario and serves as a benchmark for the controllers validation. In addition to the common scenarios, the simulations in Straume® also cover how wrong parameterization of the often unknown parameters affects the controllers performance.

A regular integrator backstepping controller was developed for the 2nd order system. This controller performed well during the simulation and thus a controller for the 3rd order system was developed. The 3rd order controller also performed well during simulations. The adaptive controller developed for the 2nd order system performed badly during simulations and a 3rd order adaptive controller was not developed.

The regular integrator backstepping controllers went through further testing by simulating them with Straume®. Both controllers performed well, but both experienced a constant offset between the reference value and the choke pressure. The controllers also experienced a drop in performance when the parametrization of the bulk modulus did not match the bulk modulus in Straume®. These controllers performed well enough for them to be used further as long as one is aware of their shortcomings.

Contents

Chapter 1

Introduction

This chapter covers the background for this thesis and the problem that need to be solved. This chapter also covers theory on both regular and adaptive integrator backstepping controllers and how to prove stability of systems on cascade form using cascade theory. There is a brief explanation on Straume®, a high-end multiphase well simulator provided by Kelda Drilling Controls for this thesis. The objectives that need to be solved and the approach to do so is outlined towards the end of this chapter with a brief explanation of necessary limitations. The final section in this chapter covers the structure of the rest of the report.

1.1 Background

In the field of oil drilling there is a concept known as Managed Pressure Drilling (MPD). In MPD one relies on mud pumps in combination with one or more chokes (valves) in order to maintain a certain pressure at the bottom of the well.

Figure 1.1 illustrates a simplified process flow diagram of the oil drilling process. Mud is pumped into the system though the mud pump, and the mud then flows down the drillstring and through the drill bit at the end. The mud then travels back up the annulus while carrying cuttings (stones etc. from the drilling) away from the drill bit, before it finally exits the system through the chokes. The mud also lubricates the drill bit as well as providing pressure to the surroundings at the bottom of the well.

It is crucial to maintain the correct pressure at the bottom of the well, with too high pressure the well may fracture, with too low pressure the well may collapse. A collapse refers to the walls in the annulus collapsing around the drillstring due to lack of pressure at the bottom of the well. This may cause a lot of damage to the drilling equipment, trap the drillstring so that it can not be retrieved, and may ruin the drilling operation in the given well. A well fracturing is a consequence of too high pressure in the well. This causes the walls of the well to fracture and drilling mud will escape the system through the fractures, after such a fracture, the well will become significantly more susceptible to a collapse.

The mud pump is capable of delivering a constant flow of mud into the system. This is used in combination with the chokes to adjust the pressure from the mud at the bottom of the well. The actuators driving the choke has an internal speed controller the require a speed reference in order to control the valve. This creates a model with recursive structure that has the potential to be controlled with a backstepping control design. In this thesis a controller based on backstepping will be developed and evaluated in order obtain an understanding of the possibilities of utilizing the backstepping method when controlling the choke pressure.

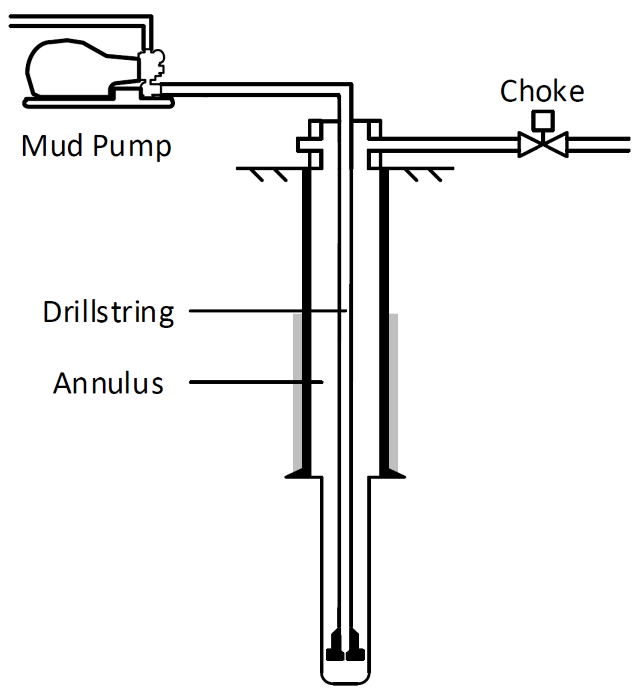

Figure 1.1: Simplified process flow diagram illustrating the flow of drilling mud through the drillstring and annulus during oil drilling

Problem Formulation

In this thesis backstepping controllers and adaptive backstepping controllers will be developed in order to control the choke pressure. The choke pressure should track a reference value and the controllers needs to achieve this by adjusting the angular velocity of the choke opening.

A design model describing the changes in choke pressure needs to be designed in order to develop controllers based on backstepping. The design model needs to contain the actuator dynamics of the choke in order relate the angular velocity of the choke to the choke pressure.

Developed controllers will go through simulations with the design model where the controllers performance is validated in common MPD scenarios. If a controller performs well in these tests, it will go through further testing in simulations with Straume®, a high-end multiphase well simulator. Straume® provides a good representation of real life drilling operations and serves as a benchmark for the controllers.

A verdict of the controllers performance will be based on the results from the simulations with Straume®.

Literature Survey

1.1.1 Integrator Backstepping

In 1989, Kokotovic and Sussmann started investigating what is now known as backstepping from passivity in Kokotovic and Sussmann (1989) and continued their work in Saberi et al. (1990). Their investigation led to results that made it possible to achieve global stabilization with full state feedback of systems on cascade form and lead to the development of integrator backstepping control design. The integrator backstepping design method is well described in Khalil (1996) and the writing in this chapter covering integrator backstepping is heavily influenced by the writing of Khalil.

In order to explain the backstepping method, first consider the following cascade form system

$$\dot{\eta} = f(\eta) + g(\eta)\xi \tag{1.1a}$$

$$\dot{\xi} = u \tag{1.1b}$$

Here $[\eta^\top, \xi] \in \mathbb{R}^{n+1}$ is the state of the system and $u \in \mathbb{R}$ is the control input. The functions $f : D \to \mathbb{R}^n$ and $g : D \to \mathbb{R}^n$ are smooth in a domain $D \subset \mathbb{R}^n$ and it is assumed that $f(\eta)$ and $g(\eta)$ are known. Equation (1.1a) can be stabilized by a smooth state feedback law $\xi = \alpha(\eta)$ with $\alpha(0) = 0$. This leads to

$$\dot{\eta} = f(\eta) + g(\eta)\alpha(\eta) \tag{1.2}$$

being asymptotically stable. It is assumed that a Lyapunov function $V(\eta)$ is known that satisfies the inequality

$$\frac{\partial V}{\partial \eta}[f(\eta) + g(\eta)\alpha(\eta)] \leq -W(\eta) \tag{1.3}$$

where $W(\eta)$ is positive definite. Since $\alpha(\eta)$ is not the exact value of ξ but rather a control input, it is assumed that there is an error between the two given by

$$z = \xi - \alpha(\eta) \tag{1.4}$$

substituting the error into the system equation yields

$$\dot{\eta} = [f(\eta) + g(\eta)\alpha(\eta)] + g(\eta)z \tag{1.5}$$

$$\dot{z} = u - \dot{\alpha} \tag{1.6}$$

Since $f(\eta)$, $g(\eta)$ and $\alpha(\eta)$ is known, $\dot{\alpha}(\eta)$ can by computed as

$$\dot{\alpha} = \frac{\partial \alpha}{\partial \eta}\dot{\eta} \tag{1.7}$$

$$\dot{\alpha} = \frac{\partial \alpha}{\partial \eta}[f(\eta) + g(\eta)\xi] \tag{1.8}$$

Taking $v = u - \dot{\alpha}$ reduces the system to the cascade connection

$$\dot{x} = [f(\eta) + g(\eta)\alpha(\eta)] + g(\eta)z \tag{1.9}$$

$$\dot{z} = v \tag{1.10}$$

which is on the same form as the original system, except now the first component has asymptotically stable origin. This will be exploited in the design of v to stabilize the overall system. Taking the Lyapunov function

$$V_c(\eta, \xi) = V(\eta) + \frac{1}{2}z^2 \tag{1.11}$$

the following is obtained

$$\dot{V}_c = \frac{\partial V}{\partial \eta}[f(\eta) + g(\eta)\alpha(\eta)] + \frac{\partial V}{\partial \eta}g(\eta)z + zv \tag{1.12}$$

$$\leq -W(\eta) + \frac{\partial V}{\partial \eta}g(\eta)z + zv \tag{1.13}$$

Choosing

$$v = -\frac{\partial V}{\partial \eta}g(\eta) - kz \tag{1.14}$$

with $k > 0$ yields

$$\dot{V}_c \leq -W(\eta) - kz^2 \tag{1.15}$$

which shows the the origin of both η and z is asymptotically stable. Since $\alpha(0) = 0$ it is also concluded that the origin of η and ξ is asymptotically stable. Substituting for v, z and $\alpha(\eta)$ givens the control input for u

$$u = \frac{\partial \alpha}{\partial x}[f(\eta) + g(\eta)\xi] - \frac{\partial V}{\partial \eta} g(\eta) - k[\xi - \alpha(\eta)] \tag{1.16}$$

If all assumptions hold globally and $V(\eta)$ is radially unbounded, the origin is globally asymptotically stable.

2nd order system example

The system

$$\dot{x}_1 = x_1^2 - x_1^3 + x_2 \tag{1.17}$$

$$\dot{x}_2 = u \tag{1.18}$$

takes the form of Equation 1.1 with $x_1 = \eta$ and $x_2 = \xi$ where x_2 is viewed as the input to $\dot{x}_1$. The input $x_2 = \alpha(x)$ will need to be designed as a feedback control in order to stabilize x_1 to the origin ($x_1 = 0$). With

$$x_2 = \alpha(x) = -x_1^2 - x_1 \tag{1.19}$$

the x_1^2 term in $\dot{x}_1$ is canceled and $\dot{x}_1$ becomes

$$\dot{x}_1 = -x_1 - x_1^3 \tag{1.20}$$

By choosing the Lyapunov function $V(x_1) = \frac{1}{2}x_1^2$, its derivative becomes

$$\dot{V} = -x_1^2 - x_1^4 \leq -x_1^2 = -W(x_1), \quad \forall x_1 \in \mathbb{R} \tag{1.21}$$

Hence, the origin of $\dot{x}_1 = -x_1 - x_1^3$ is globally exponentially stable. In order to perform the backstep, a change of variables are introduced and gives

$$e_2 = x_2 - \alpha(x_1) = x_2 + x_1 + x_1^2 \tag{1.22}$$

substituting this change into the original system yields

$$\dot{x}_1 = x_1^2 - x_1^3 + e_2 + \alpha(x) \tag{1.23}$$

$$\dot{e}_2 = u - \dot{\alpha}(x) \tag{1.24}$$

Where the derivative of $\alpha(x)$ is given by

$$\dot{\alpha}(x) = \frac{\partial \alpha}{\partial x}\dot{x} = (-1 - 2x_1)(-x_1 - x_1^3 + e_2) \tag{1.25}$$

The final system after the backstepping procedure becomes

$$\dot{x}_1 = -x_1 - x_1^3 + e_2 \tag{1.26}$$

$$\dot{e}_2 = u - (-1 - 2x_1)(-x_1 - x_1^3 + e_2) \tag{1.27}$$

In order to find the control input u, the following Lyapunov function is chosen

$$V_c(x) = V(x) + \frac{1}{2}e_2^2 \tag{1.28}$$

$$V_c(x) = \frac{1}{2}x_1^2 + \frac{1}{2}e_2^2 \tag{1.29}$$

and its derivative becomes

$$\dot{V}_c(x) = x_1(-x_1 - x_1^3 + e_2) + e_2[u + (1 + 2x_1)(-x_1 - x_1^3 + e_2)] \tag{1.30}$$

$$= -x_1^2 - x_1^4 + e_2[x_1 + (1 + 2x_1)(-x_1 - x_1^3 + e_2) + u] \tag{1.31}$$

Taking the feedback control law

$$u = -x_1 - (1 + 2x_1)(-x_1 - x_1^3 + e_2) - e_2 \tag{1.32}$$

yields

$$\dot{V}_c(x) = -x_1^2 - x_1^4 - e_2^2 \tag{1.33}$$

Hence, the system with the chosen feedback control law u is globally asymptotically stable.

3rd order system example

The following third-order system

$$\dot{x}_1 = x_1^2 - x_1^3 + x_2 \tag{1.34}$$

$$\dot{x}_2 = x_3 \tag{1.35}$$

$$\dot{x}_3 = u \tag{1.36}$$

is an extension from the previous example with an additional integrator. From the earlier example, the second-order system

$$\dot{x}_1 = x_1^2 - x_1^3 + x_2 \tag{1.37}$$

$$\dot{x}_2 = x_3 \tag{1.38}$$

had the feedback control input

$$u = -x_1 - (1 + 2x_1)(-x_1 - x_1^3 + e_2) - (x_2 + x_1 + x_1^2) \triangleq \alpha(x_1, x_2) \tag{1.39}$$

and the Lyapunov function

$$V(x_1, x_2) = \frac{1}{2}x_1^2 + \frac{1}{2}(x_2 + x_1 + x_1^2)^2 \tag{1.40}$$

To backstep further in this example, the change of variables happens to x_3.

$$e_3 = x_3 - \alpha(x_1, x_2) \tag{1.41}$$

Substituting the change of variables into the original system yields

$$\dot{x}_1 = x_1^2 - x_1^3 + x_2 \tag{1.42}$$

$$\dot{x}_2 = \alpha(x_1, x_2) + e_3 \tag{1.43}$$

$$\dot{x}_3 = u - \frac{\partial\alpha}{\partial x_1}(x_1^2 - x_1^3 + x_2) - \frac{\partial\alpha}{\partial x_2}(\alpha + e_3) \tag{1.44}$$

Using the composite Lyapunov function $V_c = V + \frac{1}{2}e^2$ gives

$$\dot{V}_c = \frac{\partial V}{\partial x_1}(x_1^2 - x_1^3 + x_2) + \frac{\partial V}{\partial x_2}(e_3 + \alpha) \tag{1.45}$$

$$+ e_3[u - \frac{\partial\alpha}{\partial x_1}(x_1^2 - x_1^3 + x_2) - \frac{\partial\alpha}{\partial x_2}(\alpha + e_3)] \tag{1.46}$$

$$= -x_1^2 - x_1^4 - (x_2 + x_1 + x_1^2)^2 \tag{1.47}$$

$$+ e_3[u - \frac{\partial\alpha}{\partial x_1}(x_1^2 - x_1^3 + x_2) - \frac{\partial\alpha}{\partial x_2}(\alpha + e_3) + \frac{\partial V}{\partial x_2}] \tag{1.48}$$

Taking the feedback control u

$$u = \frac{\partial\alpha}{\partial x_1}(x_1^2 - x_1^3 + x_2) + \frac{\partial\alpha}{\partial x_2}(\alpha + e_3) - \frac{\partial V}{\partial x_2} - e_3 \tag{1.49}$$

yields

$$\dot{V}_c = -x_1^2 - x_1^4 - (x_2 + x_1 x_1^2)^2 - e_3^2 \tag{1.50}$$

and the origin of the system is globally asymptotically stable with the chosen control input u. Note that the controller for the 3rd order system is significantly larger and more complex than the controller of the 2nd order system.

1.1.2 Adaptive Integrator Backstepping

A procedure for the development of an adaptive backstepping controller with tracking is covered by Zhou and Wen (2008). The method relies on parameter estimations of the unknown parameters in combination with a control law. This is necessary since any control laws developed for the controller can not rely on unknown parameters, and therefore the control laws make use of the estimated parameters instead. This method does not guarantee that the estimated parameter converges to the correct values, but it ensures boundedness of the closed loop states and asymptotically tracking of the reference signal. The idea is best illustrated by an example, and the second order system in Equation 1.51 will be used for the development of an adaptive backstepping controller. The development of the adaptive backstepping controller is based on the steps presented in Zhou and Wen (2008) and the results in this section are fairly similar to the results in this book.

$$\dot{x}_1 = x_2 + \phi_1^T(x_1)\theta \tag{1.51a}$$

$$\dot{x}_2 = u + \phi_2^T(x_1, x_2)\theta \tag{1.51b}$$

Here $\theta \in \mathbb{R}$ is a vector containing the unknown constant parameters in the system. $\phi_1^T(x_1)$, $\phi_2^T(x_1, x_2) \in \mathbb{R}$ are known nonlinear functions. In order to develop a controller that ensures asymptotic tracking of x_r by x_1, it is required that the first and second derivative of x_r is available and are piecewise continuous and bounded.

In order to to perform the backstepping procedure, the following virtual states and their derivatives are introduced where $\phi_1^T = \phi_1^T(x_1)$ and $\phi_2^T = \phi_2^T(x_1, x_2)$.

$$e_1 = x_1 - x_r \qquad e_2 = x_2 - \alpha_1 - \dot{x}_r \tag{1.52a}$$

$$\dot{e}_1 = e_2 + \alpha_1 + \dot{x}_r + \phi_1^T\theta - \dot{x}_r \qquad \dot{e}_2 = u + \phi_2^T\theta - \dot{\alpha}_1 - \ddot{x}_r \tag{1.52b}$$

$$\dot{e}_1 = e_2 + \alpha_1 + \phi_1^T\theta \qquad \dot{e}_2 = u + \phi_2^T\theta - \frac{\partial\alpha_1}{\partial x_1}\left(x_2 + \phi_1^T\theta\right) - \frac{\partial\alpha_1}{\partial x_r}\dot{x}_r - \ddot{x}_r \tag{1.52c}$$

Taking the Lyapunov function $V_1 = \frac{1}{2}e_1^2$ yields

$$\dot{V}_1 = e_1\left(e_2 + \alpha_1 + \phi_1^T\theta\right) \tag{1.53}$$

If θ was known the control law α could be chosen as

$$\alpha_1 = -k_1 e_1 - \phi_1^T\theta \tag{1.54}$$

Since θ is an unknown parameter, the control law α_1 cannot be chosen as in Equation 1.54 since the control law only can consist of known parameter. By replacing the unknown parameter θ with its estimated value $\hat{\theta}$ yields

$$\alpha_1 = -k_1 e_1 - \phi_1^T\hat{\theta}_1 \tag{1.55}$$

Inserting the control law from Equation 1.55 into Equation 1.53 yields

$$\dot{V}_1 = -k_1 e_1^2 + e_1 e_2 + e_1\phi_1^T\left(\theta - \hat{\theta}_1\right) \tag{1.56}$$

$$= -k_1 e_1^2 + e_1 e_2 + e_1\phi_1^T\tilde{\theta}_1 \tag{1.57}$$

where $\tilde{\theta}_1 = \theta - \hat{\theta}_1$ is the error between the real and estimated parameters. In order to deal with the estimation error $\tilde{\theta}$ the Lyapunov function $V_2 = V_1 + \frac{1}{2}\tilde{\theta}_1^T\Gamma^{-1}\tilde{\theta}_1$ is chosen and its derivative yields

$$\dot{V}_2 = -k_1 e_1^2 + e_1 e_2 + e_1\phi_1^T\tilde{\theta}_1 + \tilde{\theta}_1^T\Gamma^{-1}\dot{\tilde{\theta}}_1 \tag{1.58}$$

$$= -k_1 e_1^2 + e_1 e_2 + \tilde{\theta}_1^T\left(\phi_1 e_1 + \Gamma^{-1}\dot{\tilde{\theta}}_1\right) \tag{1.59}$$

Here $\dot{\tilde{\theta}}_1$ is the update law for the parameter estimation. Choosing the update law as

$$\dot{\tilde{\theta}}_1 = -\Gamma\phi e_1 \tag{1.60}$$

yields

$$\dot{V}_2 = -k_1 e_1^2 + e_1 e_2 \tag{1.61}$$

In order to perform the backstepping procedure, the Lyapunov function $V_3 = V_2 + \frac{1}{2} e_2^2$ is chosen and its derivative yields

$$\dot{V}_3 = -k_1 e_1^2 + e_2 \left(e_1 + u + \phi_2^T \theta - \frac{\partial \alpha_1}{\partial x_1} (x_2 + \phi_1^T \theta) - \frac{\partial \alpha_1}{\partial \hat{\theta}} \dot{\hat{\theta}}_1 - \frac{\partial \alpha_1}{\partial x_r} \dot{x}_r - \ddot{x}_r \right) \tag{1.62}$$

$$= -k_1 e_1^2 + e_2 \left(e_1 + u - \frac{\partial \alpha_1}{\partial x_1} x_2 + \left(\phi_2 - \frac{\partial \alpha_1}{\partial x_1} \phi_1 \right)^T \theta + \frac{\partial \alpha_1}{\partial \hat{\theta}} \Gamma \phi_1 e_1 - \frac{\partial \alpha_1}{\partial x_r} \dot{x}_r - \ddot{x}_r \right) \tag{1.63}$$

Taking the controller u as

$$u = -e_1 - k_2 e_2 + \frac{\partial \alpha_1}{\partial x_1} x_2 - \left(\phi_2 - \frac{\partial \alpha_1}{\partial x_1} \phi_1 \right)^T \hat{\theta}_2 - \frac{\partial \alpha_1}{\partial \hat{\theta}} \Gamma \phi_1 e_1 + \frac{\partial \alpha_1}{\partial x_r} \dot{x}_r + \ddot{x}_r \tag{1.64}$$

yields

$$\dot{V}_3 = -k_1 e_1^2 - k_2 e_2^2 - e_2 \left(\phi_2 - \frac{\partial \alpha_1}{\partial x_1} \phi_1 \right)^T (\theta - \hat{\theta}_2) \tag{1.65}$$

$$= -k_1 e_1^2 - k_2 e_2^2 - e_2 \left(\phi_2 - \frac{\partial \alpha_1}{\partial x_1} \phi_1 \right)^T \tilde{\theta}_2 \tag{1.66}$$

To deal with the estimation error $\tilde{\theta}_2$ the Lyapunov function $V_4 = V_3 + \frac{1}{2} \tilde{\theta}_2^T \Gamma^{-1} \tilde{\theta}_2$ is chosen and its derivative yields

$$\dot{V}_4 = -k_1 e_1^2 - k_2 e_2^2 - e_2 \left(\phi_2 - \frac{\partial \alpha_1}{\partial x_1} \phi_1 \right)^T \tilde{\theta}_2 + \tilde{\theta}_2^T \Gamma^{-1} \dot{\tilde{\theta}}_2 \tag{1.67}$$

$$= -k_1 e_1^2 - k_2 e_2^2 + \tilde{\theta}_2^T (- \left(\phi_2 - \frac{\partial \alpha_1}{\partial x_1} \phi_1 \right) e_2 + \Gamma^{-1} \dot{\tilde{\theta}}_2) \tag{1.68}$$

$$\tag{1.69}$$

The update law $\dot{\hat{\theta}}_2$ can be taken as

$$\dot{\hat{\theta}}_2 = \Gamma \left(\phi_2 - \frac{\partial \alpha_1}{\partial x_1} \phi_1 \right) e_2 \tag{1.70}$$

which yields

$$\dot{V}_4 = -k_1 e_1^2 - k_2 e_2^2 \tag{1.71}$$

Note that the estimation errors $\tilde{\theta}_1$ and $\tilde{\theta}_2$ are removed from the Lyapunov function when the update laws are introduced, meaning that there is no guarantee that the estimation errors becomes 0 as $t \to \infty$. However, $e_1, e_2 \to 0$ as $t \to \infty$ and asymptotic tracking is achieved. Note that the system is over-parametrized since there are two estimations for the same parameter. This issue can be solved with the use of tuning functions which is presented further in Zhou and Wen (2008). Tuning functions are not necessary if the developed controller is not over parametrized, and tuning functions will therefore not be covered in this thesis.

1.1.3 Cascade Theory

From Lamnabhi-Lagarrigue et al. (2005), a system on cascade form as in Equation (1.72) can be proved stable if it satisfies the Lemma 1

$$\Sigma_1 : \dot{x}_1 = f_1(t, x_1) + g(t, x_1, x_2)x_2 \tag{1.72a}$$

$$\Sigma_2 : \dot{x}_2 = f_2(t, x_2) \tag{1.72b}$$

Lemma 1 **(UGAS + UGAS + UGB ⇒ UGAS)** *The cascade in Equation (1.72) is UGAS if and only if Equation (1.72b) and $\Sigma_{1_\circ} : \dot{x}_1 = f_1(t, x_1)$ are UGAS and the solutions of Equation (1.72) are UGB.*

Theorem 1 defines stability criteria for time-varying cascade systems

Theorem 1 *Let Assumption 1 hold and suppose that the trajectory of Equation (1.72b) is UGB. If moreover, Assumptions 1 - 3 are satisfied, then the solutions $x(t; t_\circ, x_\circ)$ of the system in Equation (1.72) are UGB. If furthermore, the origin of Equation (1.72b) is UGAS, then so is the origin of the cascade in Equation (1.72).*

Assumption 1 *The system $\Sigma_{1_\circ} : \dot{x}_1 = f_1(t, x_1)$ is UGAS*

Assumption 2 *There exist constants $c_1, c_2, \eta > 0$ and a Lyapunov function $V(t, x_1)$ for $\Sigma_{1_\circ} : \dot{x}_1 = f_1(t, x_1)$ such that $V : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is positive definite, radially unbounded, $\dot{V}(t, x_1) \leq 0$, and*

$$\left|\frac{\partial V}{\partial x_1}\right||x_1| \leq c_1 V(t, x_1) \qquad \forall |x_1| \geq \eta \tag{1.73}$$

$$\left|\frac{\partial V}{\partial x_1}\right| \leq c_2 \qquad \forall |x_1| \leq \eta \tag{1.74}$$

Assumption 3 *There exists two continuous functions* $\theta_1, \theta_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, *such that* $g(t, x_1, x_2)$ *satisfies*

$$|g(t, x_1, x_2)| \leq \theta_1(|x_2|) + \theta_2(|x_2|)|x_1| \tag{1.75}$$

Assumption 4 *There exists a class* $\mathcal{K}$ *function* $\alpha(\cdot)$ *such that, for all* $t_\circ \geq 0$, *the trajectories of the system in Equation* (1.72b) *satisfy*

$$\int_{t_0}^{\infty} |x_2(t; t_0, x_2(t_0))| dt \leq \alpha(|x_2(t_0)|) \tag{1.76}$$

Definition 1 *A continuous function* $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *is said to belong to class* $\mathcal{K}$ *if it is strictly increasing and* $\alpha(0) = 0$. *It is said to be of class* $\mathcal{K}_\infty$ *if moreover* $\alpha(s) \to \infty$ *as* $s \to \infty$

What Remains to be Done?

There are currently no research articles related to the development of backstepping controllers to be used for any kind of choke control in MPD. On the other hand, the research on integrator backstepping and adaptive integrator backstepping control design has been thorough for the past decades and today there are well defined step by step methods in order to develop these controllers. In this thesis the well defined methods for integrator backstepping control development will be applied to a new system, namely choke pressure control.

1.2 Straume®

Straume® is a high-end multiphase well simulator with full pressure and flow dynamics developed by Kelda Drilling Controls. The simulator is capable of simulating highly accurate pressure and flow dynamics during drilling due to its distributed dynamics. Straume® separates the drillstring and annulus into several smaller volumes and utilizes this by simulating each volume with its own states. Straume® is therefore capable of simulating how changes in the system travels as waves through drillstring and annulus, rather than assuming that changes impact the whole drillstring and annulus the same way at the same time.

Kelda uses Straume® in their research and development of products as a simulator of well behavior and in this project, Straume® will serve as the baseline for the backstepping controllers when evaluating their performance.

The Straume® version provided by Kelda for this project allows for adjustment of the following input variables:

- q_p, the flow provided by the mud pump
- z_c, two parallel chokes
- p_{sbp}, Set choke pressure (if choke is disabled)
- $Echoke$, enable choke

and the following output variables:

- q_c, Choke flow
- p_d, pressure drillstring
- p_a, pressure in annulus
- q_a, flow in annulus
- q_{bit}, bit flow
- p_c, choke flow

1.3 Objectives

The main objectives of this Master's thesis are

1. To develop design models fitting for the development of integrator backstepping controllers. The design model needs to consist of the angular velocity of the choke and needs to be on cascade form.

2. Develop a reference filter that provides the backstepping controllers with the necessary derivatives of the reference value

3. Develop integrator backstepping controllers and adaptive integrator backstepping controllers based on design models for choke pressure

4. Validate the developed controllers through simulations of real life scenarios using the design models

5. Validate the developed controllers through simulations of real life scenarios using Straume®

1.4 Limitations

The Straume® version provided for this thesis take the choke position as input rather than the angular velocity. The dynamic models used for the development of the integrator backstepping controllers will therefore be used in order to provide Straume® with a position based on the controllers angular velocity output.

1.5 Approach

Due to the complexity of backstepping controllers, two design models will be developed. The design models will be of 2nd and 3rd order. The 3rd order model will give a better representation of how the angular velocity affects the choke pressure. The 2nd order model will be a simplified version of the 3rd order system and will not include the actuator dynamics of the choke.

A reference filter will be developed in order to provide the controllers with the necessary derivatives of the reference value.

An integrator backstepping controller and an adaptive integrator backstepping controller will be developed for the 2nd order system. If the controllers perform well during the simulation tests, new controllers will be developed for the 3rd order design model.

All developed controllers will first be validated in simulations with the design models. These simulations will run through common MPD scenarios. If the controller's performance is good enough, the controllers will go through further testing with Straume®. In Straume® the controllers will go through the same common MPD scenarios. In addition there will be simulations where certain parameters in the system will be different in Straume® and the controller. These simulations will determine the robustness of the controllers in terms of wrong parametrization.

Two different scenarios will be used for the validation. The first scenario consists of a series of step changes in the reference signal. The reference signal will increase by 10, 20, 30 and 40 bar and then decrease by 40, 30, 20 and 10 bar. This validates whether the controller can track step changes that vary in magnitude, for both increasing and decreasing step changes. The second scenario is a connection. This scenario is a common operation in oil drilling that is performed in order to lengthen the drillstring. In this scenario the flow q_{in} will be ramped down from 2000 $\mathrm{l\,min^{-1}}$ to 0 $\mathrm{l\,min^{-1}}$ and then ramp back up to 2000 $\mathrm{l\,min^{-1}}$ after a certain time. This validates whether the controller can handle changes in flow. In this scenario the controller is expected to maintain the choke pressure at the same value as the reference value during the flow changes. The flow into the choke and the pressure reference value will be the only variables that directly affects the choke pressure and these scenarios will therefore serve as good validation tests for the controllers.

There are two parameters in the system that often are unknown, these are the bulk modulus and the density of the fluid. The bulk modulus will change if gas leaks into the system, causing the bulk modulus to decrease. The fluid density will vary based on the actual fluid running through the system and may increase or decrease depending of the mixture of the fluid. Simulations of the two previously mentioned scenarios will therefore be performed where the bulk modulus is lower in Straume® than in the controller's parameter. The same scenarios will be simulated where the density of the fluid in Straume® is both higher and lower than in the controller's parameter.

1.6 Structure of the Report

The rest of the report is organized as follows. Chapter 2 covers the necessary preparations in order to develop integrator backstepping controllers. The development of design models that will be used in the development of the backstepping controllers is in this chapter and includes both a 2nd and 3rd order model. In addition the development of a reference filter that will provide the integrator backstepping controllers with the necessary derivatives of a reference value occurs in this chapter.

Chapter 3 covers the development of integrator backstepping controllers for the 2nd and 3rd order design model. The validation testing of the controllers when simulated with the design models in common MPD operations is also performed in this chapter and includes a discussion of the results and a decision whether the performance of the controller is good enough for further testing.

Chapter 4 covers the development of an adaptive integrator backstepping controller and are structured the same way as Chapter 3 in regards to validation testing.

In Chapter 5 the controller that passed the initial testing from Chapter 3 and Chapter 4 goes through simulations with Straume®, providing a more life like situation for the controllers. The controllers will go through the same common MPD scenarios as in the earlier testing, but in addition simulations with bad parametrization will be simulated in order to determine the robustness of the controllers. Straume® will serve as a benchmark for the controllers final validation.

Chapter 6 covers the summary and conclusions for this thesis, as well as a discussion regarding the results for the developed controllers. This chapter ends with recommendations for further work regarding the development of integrator backstepping controllers for choke pressure.

Chapter 2

Design Models and Reference Filter

2.1 Preparing the Design Models

A dynamic model of the behavior of pressure p_c is needed in order to develop backstepping controllers. In this thesis the backstepping controllers will use the choke angular velocity in order track a reference signal with the choke pressure p_c.

The choke pressure dynamics is given by

$$\dot{p}_c = \frac{\beta}{V}(q_{in} - q_c) \tag{2.1}$$

Where p_c is the pressure over the choke, β is the bulk modulus of the fluid, V is the volume of the annulus, q_{in} is the flow into the the choke and q_c is the flow out of the choke. q_c is given by

$$q_c = K_c\sqrt{\frac{2}{\rho}\left(p_c - p_{co}\right)}g(z) \tag{2.2}$$

Here K_c is choke gain, ρ is the density of the fluid running through the choke, p_c is the pressure in front of the choke while p_{co} is the pressure after the choke. $g(z)$ is a mapping function taking the choke position z and maps it to the area opening in the choke given in %. Inserting

Equation (2.2) into Equation (2.1) yields

$$\dot{p}_c = \frac{\beta}{V}\left(q_{in} - K_c\sqrt{\frac{2}{\rho}(p_c - p_{co})}g(z)\right) \tag{2.3}$$

$$= \frac{\beta}{V}q_{in} - \frac{\beta}{V}K_c\sqrt{\frac{2}{\rho}(p_c - p_{co})}g(z) \tag{2.4}$$

The pressure dynamics over the choke p_c are now given as a differential equation with the choke position z as an input.

In order to use angular velocity as the controller input, the choke actuator dynamics need to be added to the dynamic equation for the system.

$$\dot{p}_c = \frac{\beta}{V}q_{in} - \frac{\beta}{V}K_c\sqrt{\frac{2}{\rho}(p_c - p_{co})}g(z) \tag{2.5a}$$

$$\dot{z} = \omega \tag{2.5b}$$

$$\dot{\omega} = \frac{1}{\tau_\omega}(-\omega + \text{sat}(\omega_u)) \tag{2.5c}$$

Here ω is the angular velocity of the choke position, ω_u is the angular velocity input (i.e. the controller output in which to control the system) and τ_ω is a time constant. The actuator dynamics are represented as a first order linear filter in order to model the delay from the angular velocity input to the actual choke position.

The backstepping development method tends to create large complicated controllers that increase significantly in size and complexity as the order of the system increases. Because of this, a 2nd order order system will also be used in this thesis in order to have a simpler system in which to develop a controller for. The simplified system can be written as

$$\dot{p}_c = \frac{\beta}{V}q_{in} - \frac{\beta}{V}K_c\sqrt{\frac{2}{\rho}(p_c - p_{co})}g(z) \tag{2.6a}$$

$$\dot{z} = \text{sat}(\omega_u) \tag{2.6b}$$

Here the dynamics in the choke are excluded from the system dynamics and angular velocity ω

is set to be the angular velocity output from the controller. While this simplification provides a mathematically correct relation between the angular velocity and choke position, excluding the dynamics within the choke might lead to a lesser performing controller when it is developed on the 2nd order system rather than the 3rd order.

All variables and constants in the 2nd and 3rd order dynamic equations are listed in Table 2.1. p_c is the choke pressure and depends on the flow through the choke. By adjusting this pressure it is possible to control the pressure at the bottom of the well and thus providing the right pressure for drilling. p_r is a reference signal for the wanted choke pressure, this variable is not a part of the original dynamic equations, but will be added as a part of the development of a backstepping controller in order for the system to be capable of tracking this signal. p_{co} is the pressure on the output side of the choke, which is typically at 1 bar. q_{in} is the flow into the choke, this flow is controlled by the mud pumps which provides a steady flow through the drillstring, annulus and through the choke. β is the bulk modulus of the fluid. The bulk modulus of the drill mud is usually known, however, by the time the fluid has reached the choke it consists of a combination of drilling mud, cuttings from the drilling and in some cases even gas from the surroundings. This makes it harder to correctly define an accurate value for the bulk modulus β. ρ is the density of the fluid in the system. As with the bulk modulus β, accurately defining ρ might be challenging due to the mixture in the fluid by the time it reaches the choke. K_c is the choke gain. z is the choke position and operates between the range $0-1$ (representing $0-100\%$) and is also physically limited to stay within this range. $g(z)$ is a mapping function for the choke opening, taking the choke position z as input and maps the value of z to a corresponding percentage area opening inside the choke, also given in the range $0-1$ (again representing $0-100\%$). ω is the choke angular velocity and represents the velocity of which z is changing, due to the physical limitations of the choke, ω is physically limited between to $\pm 50\,\%\,\mathrm{s}^{-1}$. ω_u is the input angular velocity from the controller that will be developed in this thesis. Due to the physical limitations in the choke, ω_u is saturated in the dynamic equations in order to ensure that the physical constraints are maintained. τ_ω is the time constant for the choke's actuator dynamics.

The development of backstepping controllers quickly increases in size during the design procedure. In order for easier notation the models can be rewritten to make it easier to keep

p_c	Choke pressure	Pa
p_r	Choke pressure reference	Pa
p_{co}	Choke output pressure	Pa
q_{in}	Choke in-flow	$\frac{m^3}{s}$
β	bulk modulus	Pa
V	Volume of annulus	m^3
ρ	Density of fluid	$\frac{kg}{m^3}$
K_c	Choke gain constant	−
z	Choke position	%
$g(z)$	Mapping function	%
ω	Choke angular velocity	$\frac{\%}{s}$
ω_u	Choke angular velocity input	$\frac{\%}{s}$
τ_ω	actuator time constant	−

Table 2.1: Variable and constant table for the 2nd and 3rd order dynamic models

track of the essential parts of the controllers development. The 2nd order model from Equation (2.6) can be rewritten on the form of Equation (2.7)

$$\dot{p}_c = f_1 + g_1 g(z) \tag{2.7a}$$

$$\dot{z} = \text{sat}(\omega_u) \tag{2.7b}$$

where

$$f_1 = \frac{\beta}{V} q_{in} \tag{2.8a}$$

$$g_1 = -\frac{\beta}{V} K_c \sqrt{\frac{2}{\rho}(p_c - p_{co})} \tag{2.8b}$$

The 3rd order design model which is given by Equation (2.5) can be rewritten on the form given by Equation (2.9)

$$\dot{p}_c = f_1 + g_1 g(z) \tag{2.9a}$$

$$\dot{z} = \omega \tag{2.9b}$$

$$\dot{\omega} = f_2 + g_2 \mathrm{sat}(\omega_u) \tag{2.9c}$$

Here f_1 and g_1 is given by Equation (2.8). f_2 and g_2 is given by

$$f_2 = -\frac{\omega}{\tau_\omega} \tag{2.10a}$$

$$g_2 = \frac{1}{\tau_\omega} \tag{2.10b}$$

At this point there is a problem with the current design models. This is due to z not being linearly multiplied into a term in $\dot{p}_c$, but rather defined as an input in the mapping function $g(z)$. Figure 2.1 illustrates the functionality of $g(z)$. Here z is the position of the choke actuator and $g(z)$ is the opened area within the choke. The reason for this behavior is due to the mechanical design of the choke. The choke will be completely closed until the actuator position reaches a certain value. In addition, the area opening inside the choke is not designed in such a way that it is linear, thus causing the behavior as seen in Figure 2.1.

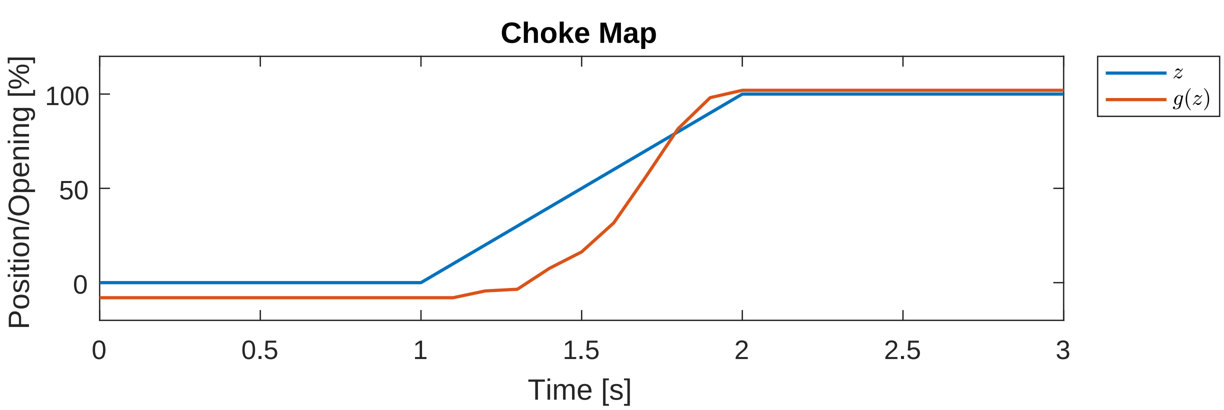

Figure 2.1: Plot describing how the mapping function $g(z)$ maps the actuator position to the actual area choke opening.

This causes problems when designing a integrator backstepping controller which becomes apparent in the following calculations. In order to perform the beckstepping procedure, there is a need to create virtual states. For the 2nd order design model (the same problem occurs the same way in both design models) there are two obvious choices for the virtual states, the first of which is

$$e_1 = p_c - p_r \qquad e_2 = g(z) - \alpha \tag{2.11a}$$

$$\dot{e}_1 = f_1 + g_1(e_2 + \alpha) - \dot{p}_r \qquad \dot{e}_2 = \frac{\partial g(z)}{\partial t} - \dot{\alpha} \tag{2.11b}$$

In this case the problem comes down to the term $\frac{\partial g(z)}{\partial t}$. It is not possible to symbolically differentiate the mapping function $g(z)$, making this a bad choice for the virtual state e_2. It could be tempting to just differentiate the term z within the mapping function, i.e. $\frac{\partial g(z)}{\partial t} = g(\dot{z}) = g(\omega_u)$. However, doing this only results in mapping the signal ω_u, which is the angular velocity of the choke, to the choke position. This will clearly not provide the correct behavior for the virtual state.

The second obvious choice would be

$$e_1 = p_c - p_r \qquad e_2 = z - \alpha \tag{2.12a}$$

$$\dot{e}_1 = f_1 + g_1 g(e_2 + \alpha) - \dot{p}_r \qquad \dot{e}_2 = \omega_u - \dot{\alpha} \tag{2.12b}$$

In this case the virtual states is defined such that it is possible to provide the controller with all the necessary signals. Taking the following Lyapunov function and its derivative yields

$$V_1 = \frac{1}{2} e_1^2 \tag{2.13}$$

$$\dot{V}_1 = e_1(f_1 + g_1 g(e_2 + \alpha) - \dot{p}_r) \tag{2.14}$$

In this case the control law α becomes

$$\alpha = -e_2 h\left(\frac{-f_1 - k_1 e_1 + \dot{p}_r}{g_1}\right) \tag{2.15}$$

Where $h(z)$ is a mapping function such that $\delta(h(z)) = z$. Assuming the mapping function $h(z)$ exists, this control law needs to eliminate the term e_2. This is not possible at the current step of the process, thus making the control law invalid.

In order to avoid adding e_2 to the control law, the derivative of the virtual state e_2 could be redefined as

$$\dot{e}_2 = f_1 + g_1(g(e_2) + g(\alpha)) - \dot{p}_r \tag{2.16}$$

which would provide the control law

$$\alpha = h\left(\frac{-f_1 - k_1 e_1 + \dot{p}_r}{g_1}\right) \tag{2.17}$$

In this case the control law could be obtainable, however, it assumes that

$$g(e_2 + \alpha) = g(e_2) + g(\alpha) \tag{2.18}$$

This is not the case, and this is illustrated in Figure 2.2. In this scenario e_2 and α both ramps up at $50\,\%\,\mathrm{s}^{-1}$ and it is clear that the behavior of $g(e_2) + g(\alpha)$ is not the same as $g(e_2 + \alpha)$. This makes sense considering that the mapping function is not linear and thus Equation 2.18 cannot hold.

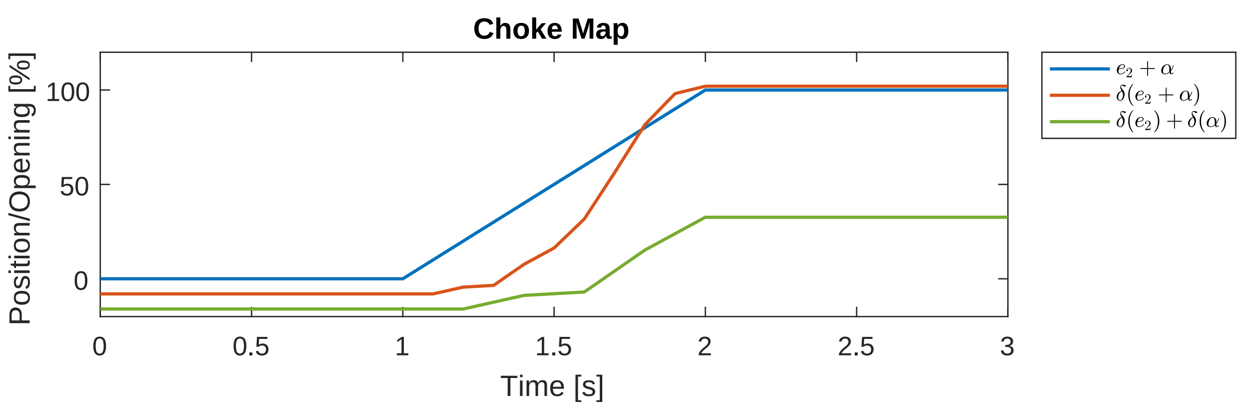

Figure 2.2: Plot illustrating how $g(e_2 + \alpha) \neq g(e_2) + g(\alpha)$ in the mapping function $g(z)$

This proves that there is not an "easy" way to develop a backstepping controller with the term $g(z)$. However, it does not necessarily mean that it is impossible to do so. Nevertheless, $g(z)$ needs to be dealt with, and an alternative solution is to avoid using $g(z)$ in the controller all together. This can be achieved by introducing the mapping function $h(z)$, which is defined such that $g(h(z)) = z$, into the design model, i.e.

$$\dot{p}_c = f_1 + g_1 g(h(z)) = f_1 + g_1 z \tag{2.19}$$

$$\dot{z} = \omega_u \tag{2.20}$$

This simplifies the task drastically as the system now has z linearly multiplied into a term in $\dot{p}_c$. This solution has two important points that needs to be addressed. The first one is that it needs to be possible to add the mapping function $h(z)$ to the system, this also goes for a system in the real world. The second one is that in order for the mapping function $h(z)$ to work well, detailed knowledge of the mapping function $g(z)$ is needed. Figure 2.3 illustrates how adding the mapping function $h(z)$ can alter the output such that $g(h(z)) = z$.

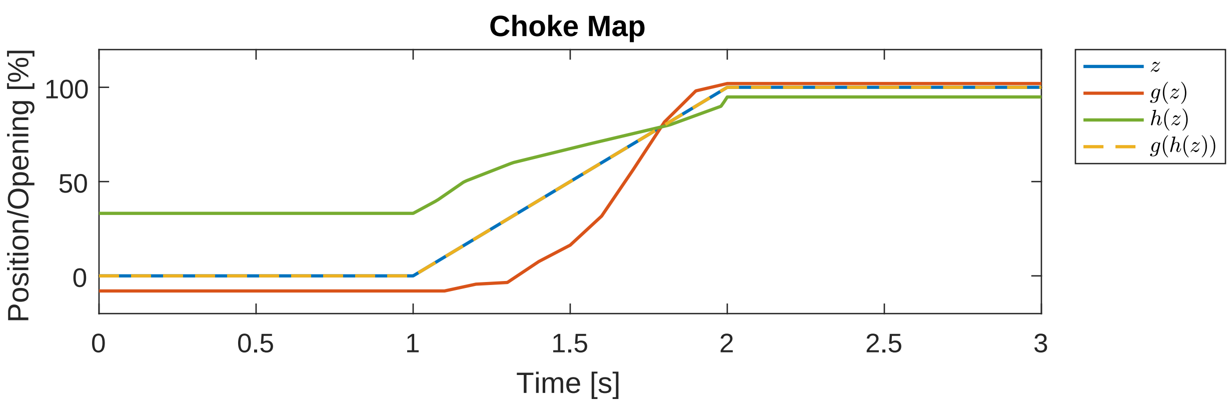

Figure 2.3: Plot illustrating how the mapping function $h(z)$ cancels out the mapping function $g(z)$ when implemented as $g(h(z))$

The new design model for the 2nd order system becomes

$$\dot{p}_c = f_1 + g_1 z \tag{2.21a}$$

$$\dot{z} = \mathrm{sat}(\omega_u) \tag{2.21b}$$

while the new design model for the 3rd order system becomes

$$\dot{p}_c = f_1 + g_1 z \tag{2.22a}$$

$$\dot{z} = \omega \tag{2.22b}$$

$$\dot{\omega} = f_2 + g_2 \text{sat}(\omega_u) \tag{2.22c}$$

For the rest of this thesis, the 2nd and 3rd order system will be referred to and worked with on the form presented in Equation (2.21) and Equation (2.22) respectively. In addition f_1, g_1, f_2, and g_2 will always be defined according to Equation (2.8) and Equation (2.10).

2.2 Tracking and Reference Filter

Tracking a reference value can be achieved by introducing an error term as in Equation 2.23, here x_r is the reference value and x_1 is the state that should track the reference value. By ensuring that the new state $e_1 \to 0$ as $t \to \infty$ tracking is achieved.

$$e_1 = x_1 - x_r \tag{2.23}$$

Consider the system in Equation 2.24. The system is on cascade form and suitable for integrator backstepping development.

$$\dot{x}_1 = x_2 \tag{2.24a}$$

$$\dot{x}_2 = u \tag{2.24b}$$

In order to perform the backstepping procedure the following virtual states and their derivatives could be used

$$e_1 = x_1 \qquad e_2 = x_2 - \alpha \tag{2.25a}$$

$$\dot{e}_1 = e_2 + \alpha \qquad e_2 = u - \dot{\alpha} \tag{2.25b}$$

Performing backstepping with these virtual states would cause the state $x_1 \rightarrow 0$ as $t \rightarrow \infty$. If the virtual states instead are defined as

$$e_1 = x_1 - x_r \qquad e_2 = x_2 - \alpha \tag{2.26a}$$

$$\dot{e}_1 = e_2 + \alpha - \dot{x}_r \qquad e_2 = u - \dot{\alpha} \tag{2.26b}$$

the system will track the reference value since $e_1 \rightarrow 0$ as $t \rightarrow \infty$.

In order to perform the backstepping procedure the Lyapunov function $V_1 = \frac{1}{2}e_1^2$ is introduced and its derivative becomes

$$\dot{V}_1 = e_1(e_2 + \alpha - \dot{x}_r) \tag{2.27}$$

taking the control law $\alpha = -k_1 e_1 + \dot{x}_r$ yields

$$\dot{V}_1 = -k_1 e_1^2 + e_1 e_2 \tag{2.28}$$

In order to perform the backstepping procedure the Lyapunov function $V_2 = V_1 + \frac{1}{2}e_2^2$ is chosen and its derivative yields

$$\dot{V}_2 = -k_1 e_1^2 + e_2(e_1 + u - \dot{\alpha}) \tag{2.29}$$

taking the controller u as in Equation 2.31

$$u = -e_1 - k_2 e_2 + \dot{\alpha} \tag{2.30}$$

$$= -e_1 - k_2 e_2 - k_1(x_2 - \dot{x}_r) + \ddot{x}_r \tag{2.31}$$

yields

$$\dot{V}_2 = -k_1 e_1^2 - k_2 e_2^2 \tag{2.32}$$

which proves that the system in Equation 2.24 is asymptotically stable and u will provide the system with a controller that tracks the reference value x_r

From this development it becomes apparent that the n first derivatives of the reference value are needed, where n is the order of the system. Since the larges design model in this thesis is of 3rd order the 3 first derivatives of the reference value p_r is needed in addition to the reference value itself, i.e. p_r, $\dot{p}_r$, $\ddot{p}_r$, and $\dddot{p}_r$. In addition, the system is not capable of tracking instantaneous step changes in the reference signal, which makes it desirable to filter the reference signal into a smooth signal the system can track. A simple reference filter can be defined as in Equation 2.33.

$$y(s)(\lambda s+1)^4 = u(s), \quad \lambda > 0, \lambda \in \mathbb{R} \tag{2.33}$$

Here $u(s)$ is the input value, $y(s)$ is the filtered output value that will be used by the controller, and $(\lambda s+1)^4$ is a 4th order filter. In this reference filter all the poles will be placed in the same position and along the negative real axis (not in the imaginary axis), providing a critically damped reference value. A critically damped reference value is preferred over an overdamped reference value since an overshoot in pressure after a reference change could cause a fracture or collapse in the well.

Expanding Equation 2.33 and transforming it into the time domain yields

$$y(s)(\lambda^4 s^4 + 4\lambda^3 s^3 + 6\lambda^2 s^2 + 4\lambda s + 1) = u(s) \tag{2.34}$$

$$y(s)s^4 = -\frac{4}{\lambda} y(s)s^3 - \frac{6}{\lambda^2} y(s)s^2 - \frac{4}{\lambda^3} y(s)s - \frac{1}{\lambda^4} y(s) + \frac{1}{\lambda^4} u(s) \tag{2.35}$$

$$\ddddot{y} = -\frac{4}{\lambda}\dddot{y} - \frac{6}{\lambda^2}\ddot{y} - \frac{4}{\lambda^3}\dot{y} - \frac{1}{\lambda^4} y + \frac{1}{\lambda^4} u \tag{2.36}$$

Equation 2.36 can be rewritten into state space form by reassigning the following variables

$$x_1 = y \tag{2.37}$$

$$x_2 = \dot{x}_1 = \dot{y} \tag{2.38}$$

$$x_3 = \dot{x}_2 = \ddot{y} \tag{2.39}$$

$$x_4 = \dot{x}_3 = \dddot{y} \tag{2.40}$$

$$\tag{2.41}$$

and the reference filter in state space form becomes

$$\dot{x}_1 = x_2 \tag{2.42a}$$

$$\dot{x}_2 = x_3 \tag{2.42b}$$

$$\dot{x}_3 = x_4 \tag{2.42c}$$

$$\dot{x}_4 = -\frac{4}{\lambda}x_4 - \frac{6}{\lambda^2}x_3 - \frac{4}{\lambda^3}x_2 - \frac{1}{\lambda^4}x_1 + \frac{1}{\lambda^4}u \tag{2.42d}$$

Reassigning the variables from the reference filter in order to obtain the variables needed for the system yields

$$p_r = x_1 \tag{2.43a}$$

$$\dot{p}_r = x_2 \tag{2.43b}$$

$$\ddot{p}_r = x_3 \tag{2.43c}$$

$$\dddot{p}_r = x_4 \tag{2.43d}$$

$$r = u \tag{2.43e}$$

Here p_r and its derivatives are the filtered reference singal needed for the backstepping controller and r is the reference signal controlled by the operator.

In Figure 2.4 the reference values p_r, $\dot{p}_r$, $\ddot{p}_r$, and $\dddot{p}_r$ as a result of a step in r are plotted. The upper plot illustrates the behavior of p_r and its derivatives during a step change. The lower plot integrates the different derivatives of p_r during the same step change and validates whether the derivatives are correct. All the integrated derivatives of p_r have the same value as p_r itself, confirming that the derivatives are indeed correct. In both plots $\lambda = 1$.

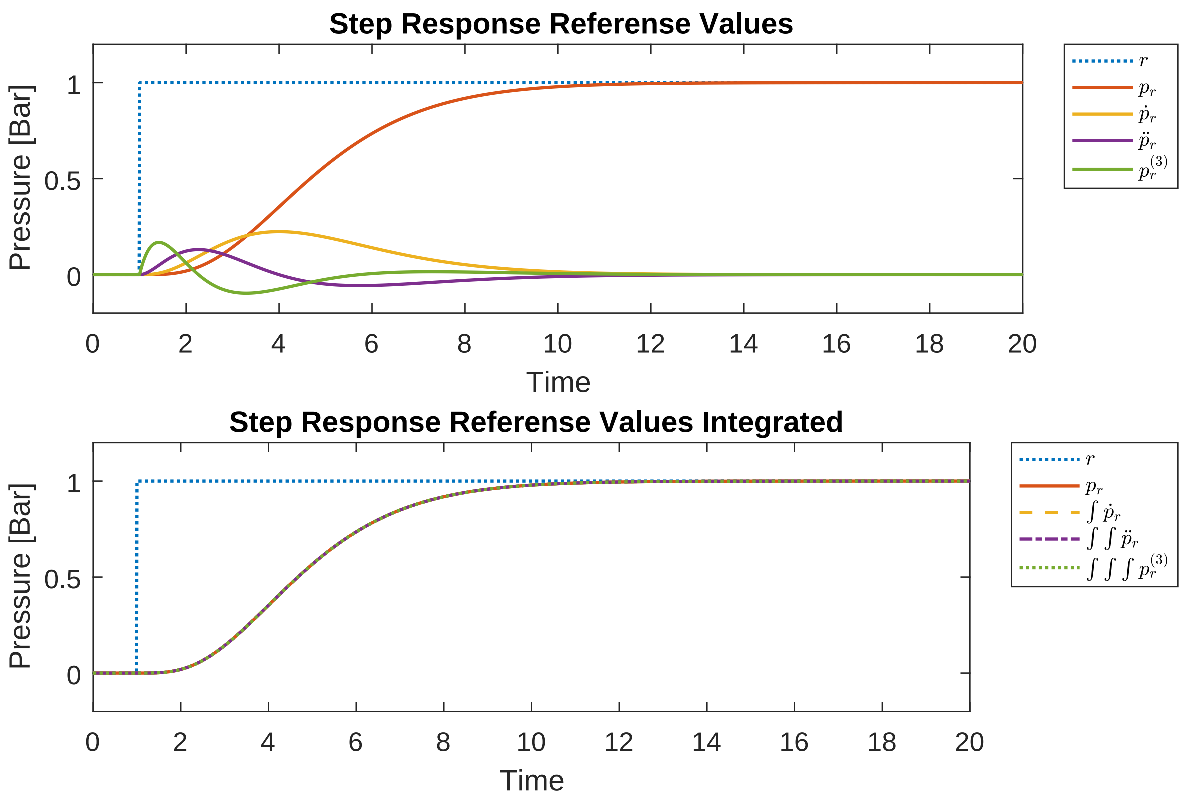

Figure 2.4: The upper plot illustrates how p_r and its derivatives responds to a step change. The lower plot verifies that the derivatives behaves correctly by integrating them in order to obtain p_r

Chapter 3

Integrator Backstepping Controller with Tracking

3.1 Integrator Backstepping with 2nd Order Design Model

Consider the the 2nd order system and note that ω_u is not saturated in this case.

$$\dot{p}_c = f_1 + g_1 z \tag{3.1a}$$

$$\dot{z} = \omega_u \tag{3.1b}$$

In this backstepping case it is assumed that the pressure p_c can be controlled through z that contains the control input ω_u. In addition, p_c is supposed to follow a reference signal p_r rather than going to 0. To achieve this, the following virtual variables are introduced.

$$e_1 = p_c - p_r \qquad e_2 = z - \alpha \tag{3.2a}$$

$$\dot{e}_1 = f_1 + g_1(e_1 + \alpha) - \dot{p}_r \qquad \dot{e}_2 = \omega_u - \dot{\alpha} \tag{3.2b}$$

Here e_1 is the difference between the pressure p_c and the reference signal p_r. By developing a controller that ensures that the error e_1 goes to 0, the pressure in the system will track the reference signal. e_2 is the difference between the control law α and the state z, ensuring that the differences between the developed control law α and the actual state z becomes 0 as e_2 goes to 0.

The first step in order to develop a backstepping controller is to ensure stability of the first state, in this case e_1. In order to create a control law for e_1, the following Lyapunov function is introduced

$$V_1 = \frac{1}{2}e_1^2 \tag{3.3}$$

and its derivative becomes

$$\dot{V}_1 = e_1(f_1 + g_1(e_2 + \alpha) - \dot{p}_r) \tag{3.4}$$

The control law α can be chosen as

$$\alpha = \frac{-f_1 + \dot{p}_r - k_1 e_1}{g_1} \tag{3.5}$$

and $\dot{V}_1$ then becomes

$$\dot{V}_1 = -k_1 e_1^2 + g_1 e_1 e_2 \tag{3.6}$$

α is not capable of handling terms that contain e_2 at this point, but e_2 will be taken care of when establishing stability for e_2. In order to control e_2 to 0 and obtain a control input, the following Lyapunov function is chosen

$$V_2 = V_1 + \frac{1}{2}e_2^2 \tag{3.7}$$

and its derivative becomes

$$\dot{V}_2 = -k_1 e_1 + e_2(g_1 e_1 + \omega_u - \dot{\alpha}) \tag{3.8}$$

Taking the control input

$$\omega_u = -k_2 e_2 - g_1 e_1 + \dot{\alpha} \tag{3.9}$$

where $\dot{\alpha}$ is given as

$$\dot{\alpha} = \frac{-\dot{f}_1 + \ddot{p}_r - k_1 \dot{e}_1}{g_1} + \frac{(-f_1 + \dot{p}_r - k_1 e_1)\dot{g}_1}{g_1^2} \tag{3.10}$$

Gives $\dot{V}_2$ the following form

$$\dot{V}_2 = -k_1 e_1^2 - k_2 e_2^2 \tag{3.11}$$

which ensures that the origin of the error system is globally asymptotically stable, and should ensure that p_c tracks the reference signal p_r. The final system with the control law α and control input becomes

$$\dot{p}_c = -k_1 e_1 + g_1 e_2 + \dot{p}_r \tag{3.12a}$$

$$\dot{z} = -k_2 e_2 - g_1 e_1 + \dot{\alpha} \tag{3.12b}$$

Simulating the system with the developed controller yields the results from Figure 3.1. During this simulation the reference signal changes from 40 bar to 20 bar. While the controller is capable of tracking this reference, it is quite clear the the controller output ω_u is not behaving well.

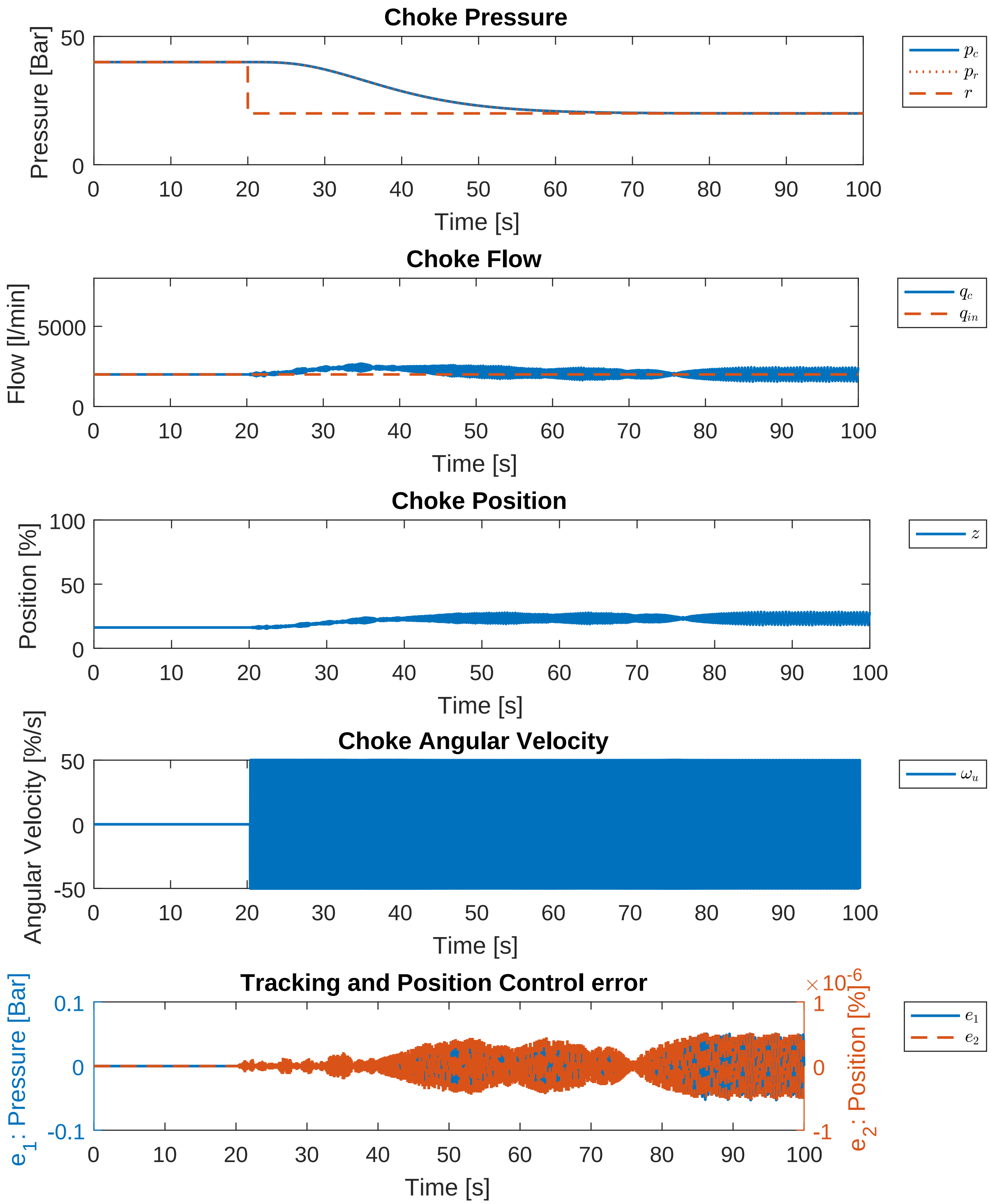

Figure 3.1: Plot illustrating how the controller behaves badly during a step change due to oscillations in ω_u

Figure 3.2 illustrates the behavior of each term in the controller $\omega_u = -k_2 e_2 - g_1 e_1 + \dot{\alpha}$ when performing the same simulation as in Figure 3.1

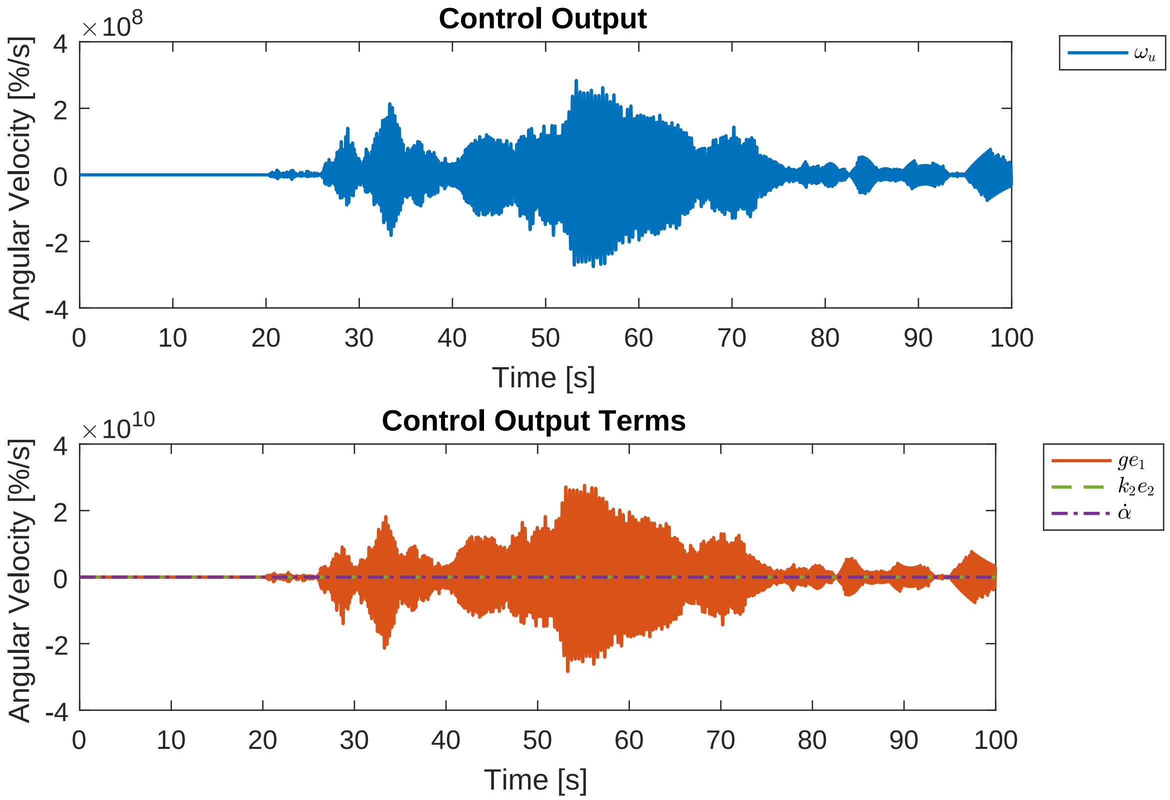

Figure 3.2: Plot comparing the controller output ω_u and each term in the controller output

It is clear that ω_u is ill conditioned due to the $g_1 e_1$ term. This term is a cancellation term in order to satisfy the stability conditions from the Lyapunov candidate, and is therefore not necessarily crucial in order to obtain a working controller. Removing this term from the controller yields.

$$\omega_u = -ke_2 + \dot{\alpha} \tag{3.13}$$

Simulating the system with the new modified controller provides the results seen in Figure 3.3. It is clear that the behavior with the new controller is better than the previous controller.

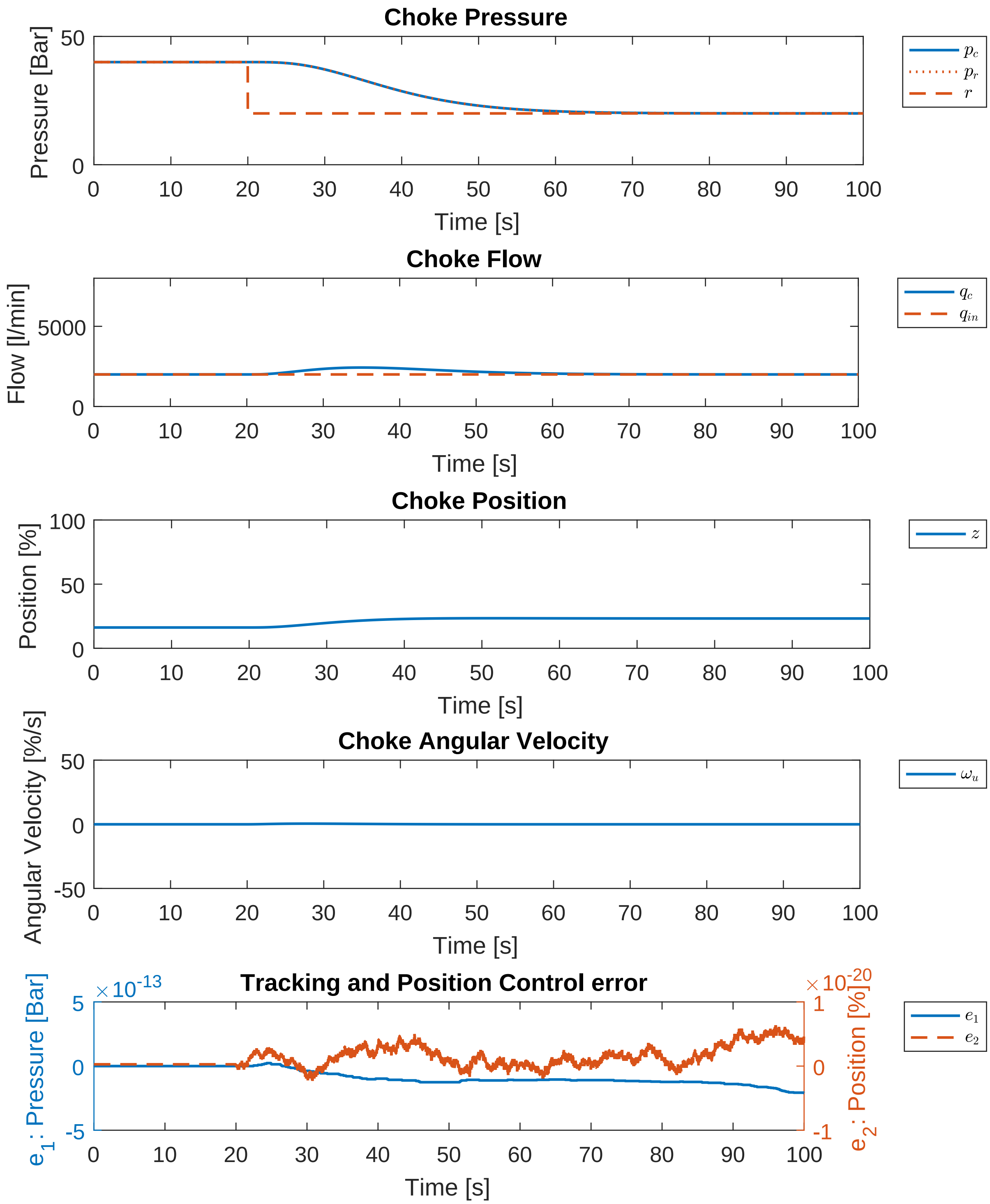

Figure 3.3: Plot illustrating the behavior of the system when the term $g_1 e_1$ is removed from the controller

With the term $g_1 e_1$ removed from the controller, $\dot{V}_2$ becomes

$$\dot{V}_2 = -k_1 e_1^2 - k_1 e_2^2 + g_1 e_1 e_2 \tag{3.14}$$

and the proof of stability from the Lyapunov candidate no longer holds since $\dot{V}_2$ is no longer negative semi definite due to the term $g_1 e_1 e_2$

By introducing the control law and controller to the virtual state equation yields

$$\dot{e}_1 = -k_1 e_1 + g_1 e_2 \tag{3.15a}$$

$$\dot{e}_2 = -k_2 e_2 \tag{3.15b}$$

and these equations are written on the cascade form which makes it possible to validate the controllers stability with cascade theory. In order for the cascade system to be stable, the system needs to satisfy the assumptions from Section 1.1.3.

For Assumption 1, $f(t, x_1) = -k_1 e_1$ needs to be UGAS. Solving $\dot{e}_1 = -k_1 e_1$ yields

$$\frac{\partial e_1}{\partial t} = -k_1 e_1 \tag{3.16}$$

$$\int \frac{1}{e_1} \partial e_2 = \int -k_1 \partial t \tag{3.17}$$

$$\ln|e_1| = -k_1 t + C \tag{3.18}$$

$$e_1 = C\mathrm{e}^{-k_1 t} \tag{3.19}$$

$$e_1(t; t_0, e_1(t_0)) = e_1(t_0)\mathrm{e}^{-k_1(t-t_0)} \tag{3.20}$$

The solution satisfies the following inequality

$$\left| e_1(t_0)\mathrm{e}^{-k_1(t-t_0)} \right| \leq \gamma_1 |e_1(t_0)| \mathrm{e}^{-\gamma_2(t-t_0)} \qquad \forall t \geq t_0 \tag{3.21}$$

With $\gamma_1 = 1$ and $\gamma_2 = k_1$, $\dot{e}_1$ is UGES and thus also UGAS and Assumption 1 is satisfied.

For Assumption 2, the Lyapunov candidate can be chosen as $V_{A2} = \frac{1}{2} e_1^2$ for the system $\dot{e}_1 = -k_1 e_1$. V_{A2} is positive definite and its derivative becomes

$$\dot{V}_{A2} = -k_1 e_1^2 \leq 0 \tag{3.22}$$

and V_{A2} satisfy the requirements for being a candidate for Assumption 2. For Assumption 2 the following equations need to be satisfied.

$$\left|\frac{\partial V_{A2}}{\partial e_1}\right| |e_1| \leq c_1 V_{A2}(t, x_1) \qquad \forall |e_1| \geq \eta \tag{3.23a}$$

$$\left|\frac{\partial V_{A2}}{\partial e_1}\right| \leq c_2 \qquad \forall |e_1| \leq \eta \tag{3.23b}$$

In this case it yields the following results

$$|e_1|^2 \leq c_1 \frac{1}{2} e_1^2 \qquad \forall |e_1| \geq \eta \tag{3.24a}$$

$$|e_1| \leq c_2 \qquad \forall |e_1| \leq \eta \tag{3.24b}$$

It is clear that (3.24a) holds for $c_1 \geq 2$ and (3.24b) holds for $c_2 \geq \eta$ and $\eta > 0$, and Assumption 2 is satisfied.

For Assumption 3 the following inequality needs to hold

$$|g(t, p_c, z)| \leq \theta_1(|z|) + \theta_2(|z|)|p_c| \tag{3.25}$$

$$\left| -\frac{\beta}{V} K_c \sqrt{\frac{2}{\rho}(p_c - p_{co})} \right| \leq \theta_1(|z|) + \theta_2(|z|)|p_c| \tag{3.26}$$

and it is known that

$$p_c \in [0, \infty) \tag{3.27a}$$

$$p_{co} \in [0, \infty) \tag{3.27b}$$

$$z \in [0, 1] \tag{3.27c}$$

$$p_c \geq p_{co} \tag{3.27d}$$

For simplification the system can be rewritten to

$$c_3 \sqrt{c_4(p_c - p_{co})} \leq \theta_1(z) + \theta_1(z) p_c \tag{3.28}$$

where $c_3 = \frac{\beta}{V} K_c > 0$ and $c_4 = \frac{2}{\rho} > 0$. Note that the absolute value of $g(t, p_c, z)$ is omitted since it in Equation (3.28) always is positive. θ_1 and θ_2 is chosen as

$$\theta_1(z) = z \tag{3.29a}$$

$$\theta_2(z) = z + c_5 \tag{3.29b}$$

It is clear that that $\theta_1(0) \leq \theta_1(z)$ and $\theta_2(0) \leq \theta_2(z)$ which gives

$$c_3 \sqrt{c_4(p_c - p_{co})} \leq c_5 p_c \tag{3.30}$$

$$\sqrt{p_c - p_{co}} \leq \frac{c_5}{c_3 \sqrt{c_4}} p_c \tag{3.31}$$

$$\sqrt{p_c - p_{co}} \leq p_c \qquad \frac{c_5}{c_3 \sqrt{c_4}} = 1 \tag{3.32}$$

It is clear that the inequality is satisfied if $p_c \geq 1$, which is suitable since normally $p_{co} \geq 1e5$ Pa

Assumption 4 requires that the following is satisfied

$$\int_{t_0}^{\infty} |e_2(t; t_0, e_2(t_0))| dt \leq \alpha_{A4}(|e_2(t_0)|) \tag{3.33}$$

$e_2(t; t_0, e_2(t_0))$ is obtained by solving $\dot{e}_2 = -k_2 e_2$

$$\frac{\partial e_2}{\partial t} = -k_2 e_2 \tag{3.34}$$

$$\int \frac{1}{e_2} \partial e_2 = \int -k_2 \partial t \tag{3.35}$$

$$\ln|e_2| = -k_2 t + C \tag{3.36}$$

$$e_2 = C\mathrm{e}^{-k_2 t} \tag{3.37}$$

$$e_2(t; t_0, e_2(t_0)) = e_2(t_0)\mathrm{e}^{-k_2(t-t_0)} \tag{3.38}$$

the origin of Equation 3.15b is then UGES and thus also UGAS. Integrating $e_2(t; t_0, e_2(t_0))$ yields

$$\int_{t_0}^{\infty} \left| e_2(t_0)\mathrm{e}^{-k_2(t-t_0)} \right| dt = \left[-\frac{e_2(t_0)}{k_2}\mathrm{e}^{-k_2(t-t_0)} \right]_{t_0}^{\infty} = \frac{e_2(t_0)}{k_2} \tag{3.39}$$

Validating the inequality requirement for Assumption 4

$$\frac{e_2(t_0)}{k_2} \leq \frac{|e_2(t_0)|}{k_2} = \alpha_{A4}(|e_2(t_0)|) \tag{3.40}$$

where $\alpha(x)_{A4} = \frac{x}{k_2}$ and is of class $\mathcal{K}$, and assumtion 4 is then satisfied.

All assumptions are satisfied and by Theorem 1 the system is UGAS. The control input

$$\omega_u = -k_2 e_2 + \dot{\alpha} \tag{3.41}$$

will ensure that p_c tracks p_r and the final system with the control law α and control input ω_u becomes

$$\dot{p}_c = -k_1 e_1 + g e_2 + \dot{p}_r \tag{3.42a}$$

$$\dot{z} = -k_2 e_2 + \dot{\alpha} \tag{3.42b}$$

3.1.1 Controller Validation 2nd Order Backstepping Controller

Figure 3.4 shows a simulation of the 2nd order integrator backstepping controller with the 2nd order dynamic model in the p_r step changes scenario. For each step change the controller manages to track the reference pressure p_r with the choke pressure p_c. By taking a look at e_1, it becomes clear that the tracking is not perfect, but the error is so small, with the biggest error being less than 4×10^{-13} Pa, that this error is not significant. The small error in e_2 indicates that the control law α does a good job in replicating the state z in order to control the pressure p_c, meaning that the developed control law α performs well in this scenario.

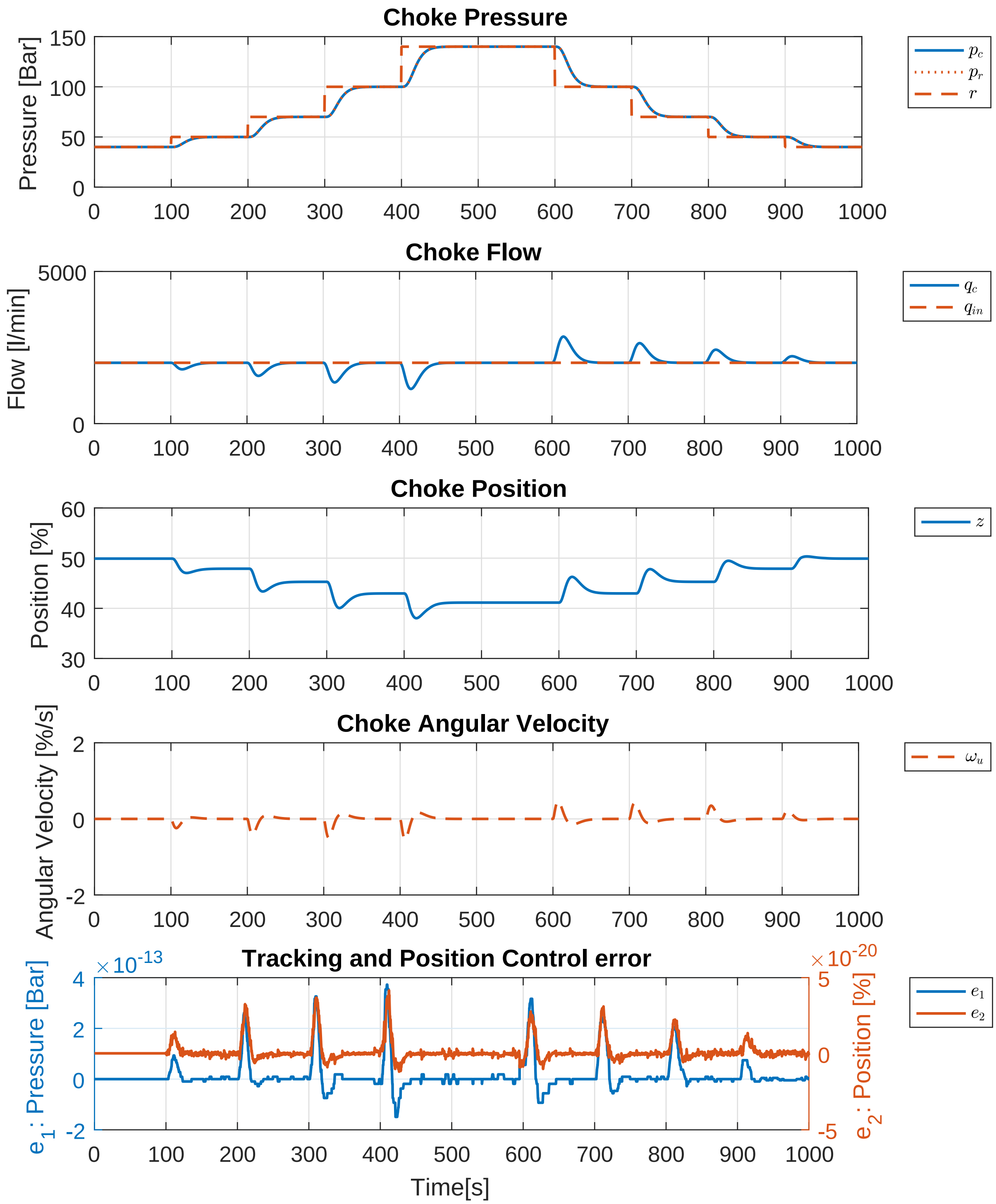

Figure 3.4: Plot results from a simulation of the 2nd order integrator backstepping controller with the 2nd order design model in the p_r step changes scenario

A simulation of the connection scenario is performed with the 2nd order integrator backstepping controller with the 2nd order dynamic model in Figure 3.5. During the whole scenario, the tracking remains perfect which can be confirmed by e_1 being 0 Pa throughout the whole simulation. The value of e_2 remains small throughout this scenario as well, confirming the good performance of the control law α.

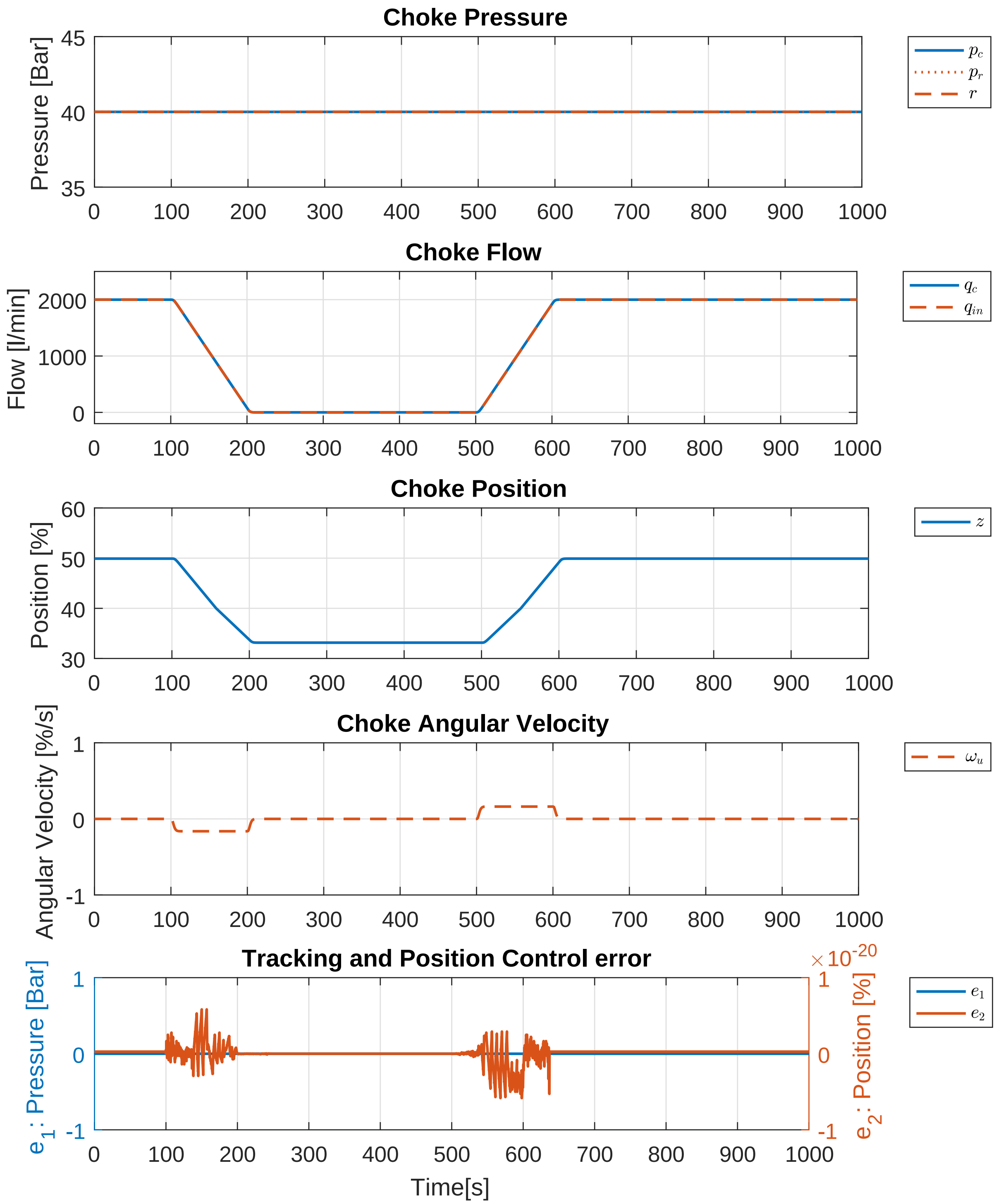

Figure 3.5: Plot results from a simulation of the 2nd order integrator backstepping controller with the 2nd order design model in the connection scenario

3.2 Integrator Backstepping with 3rd Order Design Model

Consider the 3rd order system

$$\dot{p}_c = f_1 + g_1 z \tag{3.43a}$$

$$\dot{z} = \omega \tag{3.43b}$$

$$\dot{\omega}_u = f_2 + g_2 \mathrm{sat}(\omega_u) \tag{3.43c}$$

In order to perform the backstepping procedure, the following virtual states are introduced

$$e_1 = p_c - p_r \qquad e_2 = z - \alpha_1 \qquad e_3 = \omega - \alpha_2 \tag{3.44a}$$

$$\dot{e}_1 = f + g_1(e_2 + \alpha_1) - \dot{p}_r \qquad \dot{e}_2 = e_3 + \alpha_2 - \dot{\alpha}_1 \qquad \dot{e}_3 = f_2 + g_2\omega_u - \dot{\alpha}_2 \tag{3.44b}$$

and the first Lyapunov function is chosen as

$$V_1 = \frac{1}{2}e_1^2 \tag{3.45}$$

$$\dot{V}_1 = e_1(f_1 + g_1(e_2 + \alpha_1) - \dot{p}_r) \tag{3.46}$$

taking the control law α_1 as

$$\alpha_1 = \frac{-f_1 - k_1 e_1 + \dot{p}_r}{g_1} \tag{3.47}$$

yields

$$\dot{V}_1 = -k_1 e_1^2 + g_1 e_1 e_2 \tag{3.48}$$

The next step in the backstepping procedure is introduced with the Lyapunov function V_2 and its derivative and yields

$$V_2 = V_1 + \frac{1}{2}e_2^2 \tag{3.49}$$

$$\dot{V}_2 = -k_1 e_1^2 + e_2(g_1 e_1 + e_3 + \alpha_2 - \dot{\alpha}_1) \tag{3.50}$$

taking the control law α_2 as

$$\alpha_2 = -g_1 e_1 - k_2 e_2 + \dot{\alpha}_1 \tag{3.51}$$

yields

$$\dot{V}_2 = -k_1 e_1^2 - k_2 e_2^2 + e_2 e_3 \tag{3.52}$$

The last part of the backstepping procedure is introduced with the Lypunov function V_3 and its derivative and yields

$$V_3 = V_2 + \frac{1}{2}e_3^2 \tag{3.53}$$

$$\dot{V}_3 = -k_1 e_1^2 - k_2 e_2^2 + e_3(e_2 + f_2 + g_2\omega_u - \dot{\alpha}_2) \tag{3.54}$$

Choosing the controller ω_u as

$$\omega_u = \frac{-e_2 - f_2 + \dot{\alpha}_2 - k_3 e_3}{g_2} \tag{3.55}$$

yields

$$\dot{V}_3 = -k_1 e_1^2 - k_2 e_2^2 - k_3 e_3^2 \tag{3.56}$$

where $k_1, k_2, k_3 > 0$.

As with the second order controller, the term ge_1 in $\dot{\alpha}_2$ causes problems for the system as the term is ill-conditioned. Removing this term, however, seems to provide a good controller for the system, but at the same time the proof of stability collapses. The final control laws and controller then becomes

$$\alpha_1 = \frac{-f_1 - k_1 e_1 + \dot{p}_r}{g_1} \tag{3.57a}$$

$$\alpha_2 = -k_2 e_2 + \dot{\alpha}_1 \tag{3.57b}$$

$$\omega_u = \frac{-e_2 - f_2 + \dot{\alpha}_2 - k_3 e_3}{g_2} \tag{3.57c}$$

The system can be rewritten in terms of the virtual states with the control laws α_1 and α_2 and the controller ω_u introduced to the system. The system then becomes

$$\dot{e}_1 = -k_1 e_1 + g_1 e_2 \tag{3.58a}$$

$$\dot{e}_2 = -k_2 e_2 + e_3 \tag{3.58b}$$

$$\dot{e}_3 = -k_3 e_3 - e_2 \tag{3.58c}$$

The system can be rewritten as

$$\dot{\xi}_1 = -k_1 \xi_1 + \begin{pmatrix} g_1 & 0 \end{pmatrix} \xi_2 \tag{3.59a}$$

$$\dot{\xi}_2 = \begin{pmatrix} -k_2 & 1 \\ -1 & -k_3 \end{pmatrix} \xi_2 \tag{3.59b}$$

where

$$\xi_1 = e_1 \tag{3.60a}$$

$$\xi_2 = \begin{pmatrix} e_2 \\ e_3 \end{pmatrix} \tag{3.60b}$$

$$A = \begin{pmatrix} -k_2 & 1 \\ -1 & -k_3 \end{pmatrix} \tag{3.60c}$$

The system is now written as a cascade system where ξ_2 is an input in system ξ_1. Note that the system $\dot{\xi}_1$ is the same as in Equation (3.15a), and thus, Assumption 1 - 3 already holds for this system. Theorem 1 holds if Equation 3.59b can be proven UGAS and Assumption 4 holds. For Equation 3.59b to be UGAS the eigenvalues of the matrix A needs to satisfy $\mathrm{Re}(\lambda) < 0$. The eigenvalues of A is given by

$$\det(A - \lambda I) = \begin{vmatrix} -k_2 - \lambda & 1 \\ -1 & -k_3 - \lambda \end{vmatrix} = \lambda^2 + (k_2 + k_3)\lambda + k_2 k_3 + 1 \tag{3.61}$$

solving the polynomial in Equation (3.61) yields

$$\lambda = \frac{-(k_2 + k_3) \pm \sqrt{(k_2 + k_3)^2 - 4(k_2 k_3 + 1)}}{2} \tag{3.62}$$

$$\lambda = \frac{-(k_2 + k_3) \pm \sqrt{k_2^2 + 2k_2 k_3 + k_3^2 - 4k_2 k_3 - 4}}{2} \tag{3.63}$$

$$\lambda = \frac{-(k_2 + k_3) \pm \sqrt{(k_2 - k_3)^2 - 4}}{2} \tag{3.64}$$

By analyzing Equation (3.64) the stability of the system can be validated. From the beckstepping procedure it is known that $k_2, k_3 > 0$. If $(k_2 - k_3)^2 \leq 4$, then clearly $\mathrm{Re}(\lambda) < 0$ since the values inside the square root is negative and thus imaginary. When $(k_2 - k_3)^2 > 4$, the values inside the square root are no longer imaginary. However, since the term outside the square is defined such that any increase in either k_2 or k_3 will lead to a larger negative value, while inside the square root, in order to achieve an increase in value, one of either k_2 or k_3 needs to have a low value while the other increases. Because of this behavior, the worst case scenario in terms of defining values for k_2 and k_3 (i.e. ending up with a case where $\mathrm{Re}(\lambda) < 0$ no longer holds) is when either k_2 or k_3 is so small that it is practically 0, while the other is extremely large. However, since $k_2, k_3 > 0$, the term outside the square will always be larger in magnitude that the term inside the square root, and thus, for valid choices of k_1 and k_2, $\mathrm{Re}(\lambda) < 0$ will hold. In addition, the term -4 inside the square root further strengthens this proof as it decreases the total value inside the square root. Equation 3.59b is therefor UGAS.

In order to verify whether the system satisfies Assumption 4, the solution for the system in Equation 3.59b is given by Equation 3.65

$$\xi_2 = e^{At}\xi_2(0) \tag{3.65}$$

Here e^{At} given by

$$e^{At} = f(A) = h(A) = \beta_0 I + \beta_1 A \tag{3.66}$$

where β_0 and β_1 are obtained by solving

$$h(\lambda_i) = f(\lambda_i) \tag{3.67a}$$

$$h(\lambda_i) = \beta_0 + \beta_1 \lambda_i \tag{3.67b}$$

$$f(\lambda_i) = e^{\lambda_i t} \tag{3.67c}$$

where $\lambda_i, i \in [1,2]$ are the eigenvalues of A and are obtained through Equation 3.64. Solving the equalities from Equation 3.67 yields the results from Equation 3.70 - 3.71

$$\beta_0 + \lambda_1 \beta_1 = e^{\lambda_1 t} \tag{3.68}$$

$$\beta_0 + \lambda_2 \beta_1 = e^{\lambda_2 t} \tag{3.69}$$

$$\beta_0 = -\lambda_1 \left(\frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} \right) + e^{\lambda_1 t} \tag{3.70}$$

$$\beta_1 = \frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} \tag{3.71}$$

e^{At} is then given by

$$e^{At} = \beta_0 I + \beta_1 A \tag{3.72}$$

$$= \left(-\lambda_1 \left(\frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} \right) + e^{\lambda_1 t} \right) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} \begin{pmatrix} -k_2 & 1 \\ -1 & k_3 \end{pmatrix} \tag{3.73}$$

In order to simplify further calculations, e^{At} can be be rewritten as shown in Equation

$$e^{At} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{3.74a}$$

$$a = -\lambda_1 \frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} + e^{\lambda_1 t} - k_2 \frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} \tag{3.74b}$$

$$b = \frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} \tag{3.74c}$$

$$c = -\frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} \tag{3.74d}$$

$$d = -\lambda_1 \frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} + e^{\lambda_1 t} + k_3 \frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} \tag{3.74e}$$

Setting the initial value $\xi_2(t_0) = \begin{pmatrix} \xi_{2,1}(t_0) \\ \xi_{2,2}(t_0) \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, [c_1, c_2] \in \mathbb{R}^1$ the solution of ξ_2 is given by Equation 3.75

$$\xi_2 = e^{At} \xi_2(0) \tag{3.75a}$$

$$\xi_2 = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \tag{3.75b}$$

$$\xi_{2,1} = ac_1 + bc_2 \tag{3.75c}$$

$$\xi_{2,2} = cc_1 + dc_2 \tag{3.75d}$$

In order for the system to satisfy Assumption 4, the solutions in Equations 3.75c - 3.75c needs to satisfy

$$\int_{t_0}^{\infty} |\xi_{2,1}(t; t_0, \xi_{2,1}(t_0))| dt \le \alpha_1(|\xi_{2,1}(t_0)|) \tag{3.76a}$$

$$\int_{t_0}^{\infty} |\xi_{2,2}(t; t_0, \xi_{2,2}(t_0))| dt \le \alpha_2(|\xi_{2,2}(t_0)|) \tag{3.76b}$$

Since the verification of assumption 4 relies on integration with respect to t (note that c_1 and c_4

are functions of t_0 rather than t and will behave as constants in this integration), the solutions from Equations 3.75c - 3.75c can be further simplified in order to ease the integration task. The simplification of $\xi_{2,1}$ is performed in Equation 3.77

$$\xi_{2,1} = ac_1 + bc_2 \tag{3.77a}$$
$$= \left(-\lambda_1 \frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} + e^{\lambda_1 t} - k_2 \frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1}\right) c_1 + \frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} c_2 \tag{3.77b}$$
$$= \frac{\lambda_1 c_1}{\lambda_2 - \lambda_1} e^{\lambda_1 t} + \frac{-\lambda_1 c_1}{\lambda_2 - \lambda_1} e^{\lambda_2 t} + c_1 e^{\lambda_1 t} + \frac{k_2 c_1}{\lambda_2 - \lambda_1} e^{\lambda_1 t} + \tag{3.77c}$$
$$\frac{-k_2 c_1}{\lambda_2 - \lambda_1} e^{\lambda_2 t} + \frac{-c_2}{\lambda_2 - \lambda_1} e^{\lambda_1 t} + \frac{c_2}{\lambda_2 - \lambda_1} e^{\lambda_2 t} \tag{3.77d}$$
$$= \left(\frac{\lambda_1 c_1}{\lambda_2 - \lambda_1} + c_1 + \frac{k_2 c_1}{\lambda_2 - \lambda_1} + \frac{-c_2}{\lambda_2 - \lambda_1}\right) e^{\lambda_1 t} + \tag{3.77e}$$
$$\left(\frac{-\lambda_1 c_1}{\lambda_2 - \lambda_1} + \frac{-k_2 c_1}{\lambda_2 - \lambda_1} + \frac{c_2}{\lambda_2 - \lambda_1}\right) e^{\lambda_2 t} \tag{3.77f}$$
$$= c_3 e^{\lambda_1 t} + c_4 e^{\lambda_2 t} \tag{3.77g}$$

Likewise, a simplification of $\xi_{2,2}$ is performed in Equation 3.78

$$\xi_{2,2} = cc_1 + dc_2 \tag{3.78a}$$
$$= -\frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} c_1 + \left(-\lambda_1 \frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1} + e^{\lambda_1 t} + k_3 \frac{-e^{\lambda_1 t} + e^{\lambda_2 t}}{\lambda_2 - \lambda_1}\right) c_2 \tag{3.78b}$$
$$= \frac{c_1}{\lambda_2 - \lambda_1} e^{\lambda_1 t} + \frac{-c_1}{\lambda_2 - \lambda_1} e^{\lambda_2 t} + \frac{\lambda_1 c_2}{\lambda_2 - \lambda_1} e^{\lambda_1 t} + \tag{3.78c}$$
$$\frac{-\lambda_1 c_2}{\lambda_2 - \lambda_1} e^{\lambda_2 t} + c_2 e^{\lambda_1 t} + \frac{-k_3 c_2}{\lambda_2 - \lambda_1} e^{\lambda_1 t} + \frac{k_3 c_2}{\lambda_2 - \lambda_1} e^{\lambda_2 t} \tag{3.78d}$$
$$= \left(\frac{c_1}{\lambda_2 - \lambda_1} + \frac{\lambda_1 c_2}{\lambda_2 - \lambda_1} + c_2 + \frac{-k_3 c_2}{\lambda_2 - \lambda_1}\right) e^{\lambda_1 t} + \tag{3.78e}$$
$$\left(\frac{-c_1}{\lambda_2 - \lambda_1} + \frac{-\lambda_1 c_2}{\lambda_2 - \lambda_1} + \frac{k_3 c_2}{\lambda_2 - \lambda_1}\right) e^{\lambda_2 t} \tag{3.78f}$$
$$= c_5 e^{\lambda_1 t} + c_6 e^{\lambda_2 t} \tag{3.78g}$$

Here $c_3, c_4, c_5, c_6 \in \mathbb{R}$, consists only of constant values decided by the choice of k_1 and k_2 (λ_1 and λ_2 are given by k_1 and k_2 according to Equation 3.64) and are given by

$$c_3 = c_3(c_1, c_2) = \left(\frac{\lambda_1 c_1}{\lambda_2 - \lambda_1} + c_1 + \frac{k_2 c_1}{\lambda_2 - \lambda_1} + \frac{-c_2}{\lambda_2 - \lambda_1} \right) \tag{3.79a}$$

$$c_4 = c_4(c_1, c_2) = \left(\frac{-\lambda_1 c_1}{\lambda_2 - \lambda_1} + \frac{-k_2 c_1}{\lambda_2 - \lambda_1} + \frac{c_2}{\lambda_2 - \lambda_1} \right) \tag{3.79b}$$

$$c_5 = c_5(c_1, c_2) = \left(\frac{c_1}{\lambda_2 - \lambda_1} + \frac{\lambda_1 c_2}{\lambda_2 - \lambda_1} + c_2 + \frac{-k_3 c_2}{\lambda_2 - \lambda_1} \right) \tag{3.79c}$$

$$c_6 = c_6(c_1, c_2) = \left(\frac{-c_1}{\lambda_2 - \lambda_1} + \frac{-\lambda_1 c_2}{\lambda_2 - \lambda_1} + \frac{k_3 c_2}{\lambda_2 - \lambda_1} \right) \tag{3.79d}$$

Since $e^{\lambda_1 t}, e^{\lambda_2 t} \geq 0$, then clearly

$$\int_{t_0}^{\infty} |\xi_{2,1}(t; t_0, \xi_{2,1}(t_0)| dt = \int_{t_0}^{\infty} \left| c_3 e^{\lambda_1 t} + c_4 e^{\lambda_2 t} \right| dt \leq \int_{t_0}^{\infty} \left(|c_3| e^{\lambda_1 t} + |c_4| e^{\lambda_2 t} \right) dt \tag{3.80}$$

Then integrating $\xi_{2,1}$ yields

$$\int_{t_0}^{\infty} \left(|c_3| e^{\lambda_1 t} + |c_4| e^{\lambda_2 t} \right) dt = -\frac{|c_3| e^{\lambda_1 t_0}}{\lambda_1} - \frac{|c_4| e^{\lambda_2 t_0}}{\lambda_2} \tag{3.81}$$

$$= -\frac{|c_3(c_1, c_2)| e^{\lambda_1 t_0}}{\lambda_1} - \frac{|c_4(c_1, c_2)| e^{\lambda_2 t_0}}{\lambda_2} \tag{3.82}$$

$$= -\frac{|c_3(\xi_2(t_0))| e^{\lambda_1 t_0}}{\lambda_1} - \frac{|c_4(\xi_2(t_0))| e^{\lambda_2 t_0}}{\lambda_2} \tag{3.83}$$

Likewise, for $\xi_{2,2}$

$$\int_{t_0}^{\infty} \left(|c_5| e^{\lambda_1 t} + |c_6| e^{\lambda_2 t} \right) dt = -\frac{|c_5| e^{\lambda_1 t_0}}{\lambda_1} - \frac{|c_6| e^{\lambda_2 t_0}}{\lambda_2} \tag{3.84}$$

$$= -\frac{|c_5(c_1, c_2)| e^{\lambda_1 t_0}}{\lambda_1} - \frac{|c_6(c_1, c_2)| e^{\lambda_2 t_0}}{\lambda_2} \tag{3.85}$$

$$= -\frac{|c_5(\xi_2(t_0))| e^{\lambda_1 t_0}}{\lambda_1} - \frac{|c_6(\xi_2(t_0))| e^{\lambda_2 t_0}}{\lambda_2} \tag{3.86}$$

Since the values of $\lambda_i < 0$ and $t_0 \geq 0$, the largest value of $e^{\lambda_i t}$ is 1. The class $\mathscr{K}$ functions $\alpha_i(\cdot)$ can be chosen as in Equation 3.87

$$\alpha_1(\xi_2(t_0)) = -\frac{|c_3(\xi_2(t_0))|}{\lambda_1} - \frac{|c_4(\xi_2(t_0))|}{\lambda_2} \tag{3.87a}$$

$$\alpha_2(\xi_2(t_0)) = -\frac{|c_5(\xi_2(t_0))|}{\lambda_1} - \frac{|c_6(\xi_2(t_0))|}{\lambda_2} \tag{3.87b}$$

Here the functions $c_i(\xi_2(t_0))$ are linear and since to $\lambda_i < 0$, the functions $\alpha_i(\xi_2(t_0))$ are strictly increasing. In addition $\alpha(0) = 0$ and thus satisfying the criteria for being a class $\mathcal{K}$ function. The inequalities in Equation 3.88a holds and thus Assumption 4 holds.

$$\int_{t_0}^{\infty} \left(|c_3|e^{\lambda_1 t} + |c_4|e^{\lambda_2 t}\right) dt \leq -\frac{|c_3(\xi_2(t_0))|}{\lambda_1} - \frac{|c_4(\xi_2(t_0))|}{\lambda_2} \tag{3.88a}$$

$$\int_{t_0}^{\infty} \left(|c_4|e^{\lambda_1 t} + |c_5|e^{\lambda_2 t}\right) dt \leq -\frac{|c_5(\xi_2(t_0))|}{\lambda_1} - \frac{|c_6(\xi_2(t_0))|}{\lambda_2} \tag{3.88b}$$

Theorem 1 is then satisfied the the 3rd order system is therefore UGAS.

3.2.1 Controller Validation 3rd Order Backstepping Controller

Figure 3.6 displays the results from simulating the 3rd order design model and integrator backstepping controller in the p_r step changes scenario. In this simulation, the controller does a good job in tracking the reference signal p_r with the choke pressure p_c. However, compared to the 2nd order controller from Section 3.1.1, the error e_1 is significantly larger in this case. The error e_1 is still small, with the larges peak being less than 5×10^{-3} bar, which is not even noticeable considering p_c typically having values that are greater than 1×10^5 bar. Nonetheless, this observation indicates that the extra step performed in the development of the 3rd order backstepping controller slightly impacts the performance of the controller. This is not necessarily surprising since the extra step significantly increases the number of terms and derivatives in the developed controller, and thus increases the number of possibilities for small errors in terms of offsets to appear.

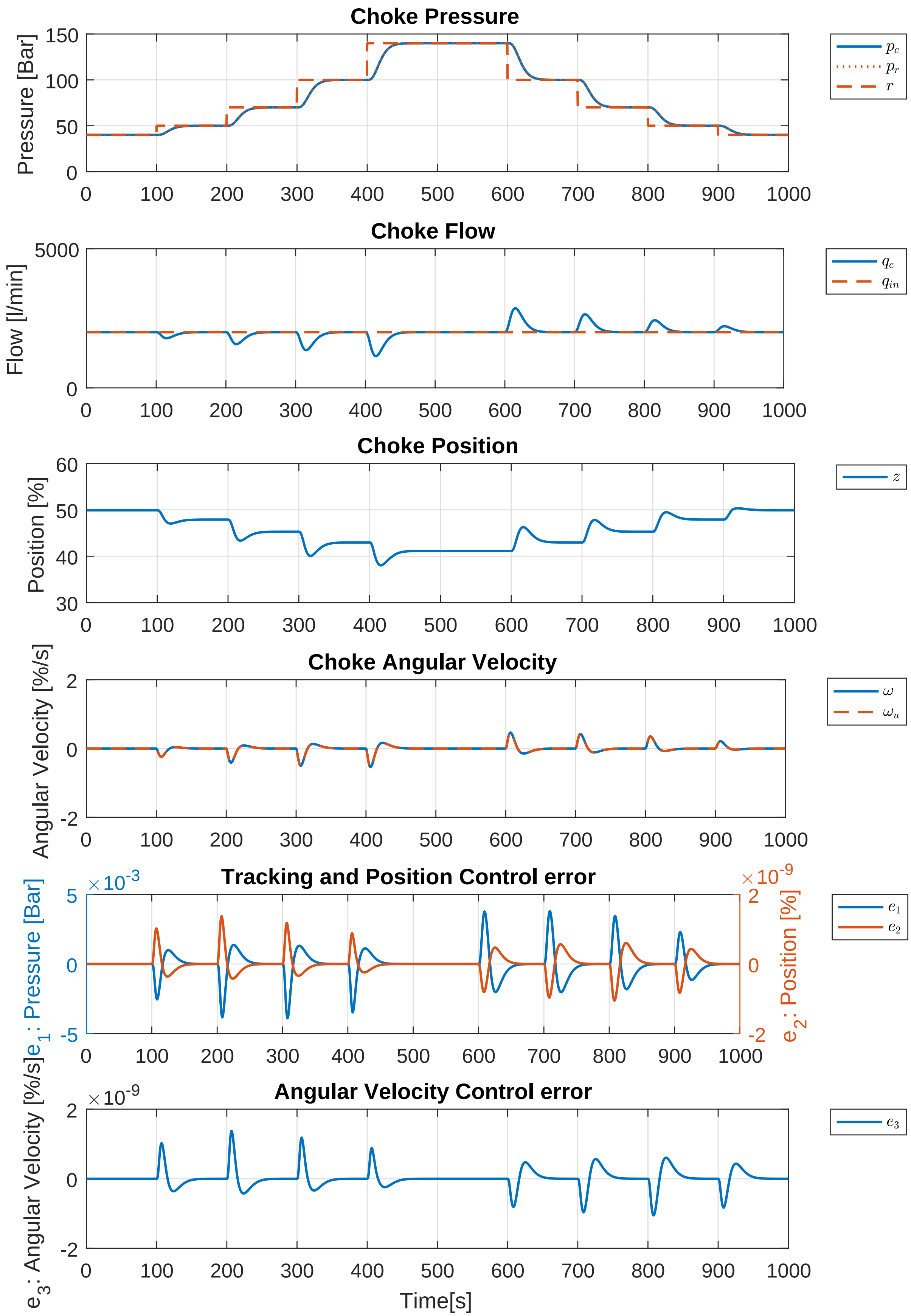

Figure 3.6: Plot results from a simulation of the 3rd order integrator backstepping controller with the 3rd order design model in the p_r step changes scenario

A simulation of the connection scenario is illustrated in Figure 3.7. In this scenario, the pressure p_c drops slightly during the ramp down of the flow q_{in}, and during the ramp up of q_{in} the pressure p_c increases slightly. The controller manages to keep the pressure p_c close to the reference value p_r. Compared to the 2nd order integrator backstepping controller which managed to keep the pressure p_c exactly equal p_r, the 3rd order controller clearly performs worse. The error between p_c and p_r is only 0.2 bar at worst which is still good.

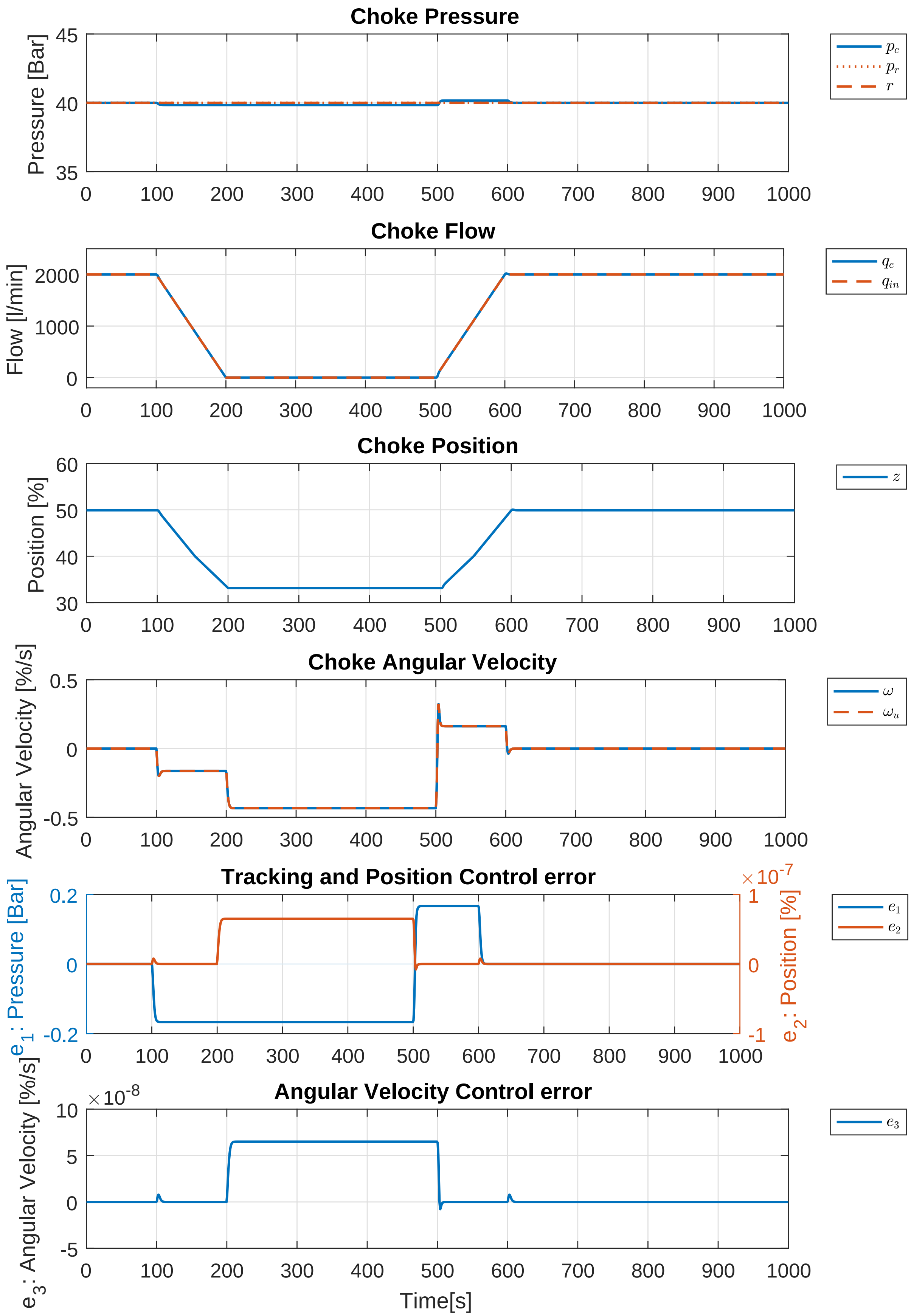

Figure 3.7: Plot results from a simulation of the 3rd order integrator backstepping controller with the 3rd order design model in the connection scenario

Chapter 4

Adaptive Integrator Backstepping Controller with Tracking

It is not uncommon in real life scenarios that some parameters are unknown in the system. This is the case for ρ and β in the system covered in this thesis. This chapter will cover the development of adaptive backstepping controllers where the parameters β and ρ are considered to be unknown.

4.1 Adaptive Controller 2nd Order System

For the second order system an adaptive controller can be developed based on the following design model

$$\dot{p}_c = \theta_1 f_3 + \theta_2 g_3 z \tag{4.1a}$$

$$\dot{z} = \omega_u \tag{4.1b}$$

where

$$f_3 = \frac{q_{in}}{V} \tag{4.2a}$$
$$g_3 = -\frac{K_c}{V}\sqrt{2(p_c - p_{co})} \tag{4.2b}$$
$$\theta_1 = \beta \tag{4.2c}$$
$$\theta_2 = \frac{\beta}{\sqrt{\rho}} \tag{4.2d}$$

Here θ_1 and θ_2 are the parameters that need to be estimated and the following error variables are introduced

$$\tilde{\theta}_1 = \hat{\theta}_1 - \theta_1 \qquad \dot{\tilde{\theta}}_1 = \dot{\hat{\theta}}_1 \tag{4.3a}$$
$$\tilde{\theta}_2 = \hat{\theta}_2 - \theta_2 \qquad \dot{\tilde{\theta}}_2 = \dot{\hat{\theta}}_2 \tag{4.3b}$$

where $\hat{\theta}_1$ and $\hat{\theta}_2$ are the estimated values of θ_1 and θ_2 respectively.

In order to perform the backstepping procedure, the following virtual states are introduced

$$e_1 = p_c - p_r \qquad e_2 = z - \alpha \tag{4.4a}$$
$$\dot{e}_1 = \theta_1 f_3 + \theta_2 g_3(e_2 + \alpha) - \dot{p}_r \qquad \dot{e}_2 = \omega_u - \dot{\alpha} \tag{4.4b}$$

The first step of the development of the adaptive backstepping controller is created with the following Lyapunov function and its derivative

$$V_1 = \frac{1}{2\theta_2} e_1^2 \tag{4.5}$$
$$\dot{V}_1 = e_1\left(\frac{\theta_1}{\theta_2} f_3 + g_3(e_2 + \alpha) - \frac{1}{\theta_2}\dot{p}_r\right) \tag{4.6}$$

Before choosing the control law α, new estimation parameters are introduced

$$\theta_3 = \frac{\theta_1}{\theta_2} = \frac{\beta}{\frac{\beta}{\sqrt{\rho}}} = \sqrt{\rho} \tag{4.7a}$$

$$\theta_4 = \frac{1}{\theta_2} = \frac{1}{\frac{\beta}{\sqrt{\rho}}} = \frac{\sqrt{\rho}}{\beta} \tag{4.7b}$$

with the corresponding estimation errors

$$\tilde{\theta}_3 = \hat{\theta}_3 - \theta_3 \qquad \dot{\tilde{\theta}}_3 = \dot{\hat{\theta}}_3 \tag{4.8a}$$

$$\tilde{\theta}_4 = \hat{\theta}_4 - \theta_4 \qquad \dot{\tilde{\theta}}_4 = \dot{\hat{\theta}}_4 \tag{4.8b}$$

The control law α is then chosen as

$$\alpha = \frac{-\hat{\theta}_3 f_3 - k_1 e_1 + \hat{\theta}_4 \dot{p}_r}{g_3} \tag{4.9}$$

and the derivative of the Lyapunov function V_1 becomes

$$\dot{V}_1 = -k_1 e_1^2 + g_3 e_1 e_2 - \tilde{\theta}_3 f_3 e_1 + \tilde{\theta}_4 \dot{p}_r e_1 \tag{4.10}$$

For the next step in the beckstepping design the following Lyapunov function is chosen

$$V_2 = V_1 + \frac{1}{2} e_2 \tag{4.11}$$

$$\dot{V}_2 = -k_1 e_1^2 - \tilde{\theta}_3 f_3 e_1 + \tilde{\theta}_4 \dot{p}_r e_1 + e_2(g_3 e_1 + \omega_u - \dot{\alpha}) \tag{4.12}$$

where $\dot{\alpha}$ is given by

$$\dot{\alpha} = \frac{-\dot{\hat{\theta}}_3 f_3 - \hat{\theta}_3 \dot{f}_3 - k_1 \dot{e}_1 + \dot{\hat{\theta}}_4 \dot{p}_r + \hat{\theta}_4 \ddot{p}_r}{g_3} - \frac{(-\hat{\theta}_3 f_3 - k_1 e_1 + \hat{\theta}_4 \dot{p}_r)\dot{g}_3}{g_3^2} \tag{4.13}$$

$$\dot{\alpha} = \frac{-\dot{\hat{\theta}}_3 f_3 - \hat{\theta}_3 \dot{f}_3 - k_1 \theta_1 f_3 - k_1 \theta_2 g_3 z + k_1 \dot{p}_r + \dot{\hat{\theta}}_4 \dot{p}_r + \hat{\theta}_4 \ddot{p}_r}{g_3} - \frac{(-\hat{\theta}_3 f_3 - k_1 e_1 + \hat{\theta}_4 \dot{p}_r)\dot{g}_3}{g_3^2} \tag{4.14}$$

The controller ω_u is then chosen as

$$\omega_u = -g_3 e_1 + \dot{\hat{\alpha}} - k_2 e_2 \tag{4.15}$$

Here $\dot{\hat{\alpha}}$ is the same as control law $\dot{\alpha}$, but instead of using the unknown parameters θ_1 and θ_2 it uses the estimated parameters $\hat{\theta}_1$ and $\hat{\theta}_2$. $\dot{\hat{\alpha}}$ is then given as

$$\dot{\hat{\alpha}} = \frac{-\dot{\hat{\theta}}_3 f_3 - \hat{\theta}_3 \dot{f}_3 - k_1 \hat{\theta}_1 f_3 - k_1 \hat{\theta}_2 g_3 z + k_1 \dot{p}_r + \dot{\hat{\theta}}_4 \dot{p}_r + \hat{\theta}_4 \ddot{p}_r}{g_3} - \frac{(-\hat{\theta}_3 f_3 - k_1 e_1 + \hat{\theta}_4 \dot{p}_r)\dot{g}_3}{g_3^2} \tag{4.16}$$

and $\dot{\tilde{\alpha}} = \dot{\hat{\alpha}} - \dot{\alpha}$ then becomes

$$\dot{\tilde{\alpha}} = -\frac{k_1 \hat{\theta}_1 f_3}{g_3} - \frac{k_1 \hat{\theta}_2 g_3 z}{g_3} + \frac{k_1 \theta_1 f_3}{g_3} + \frac{k_1 \theta_2 g_3 z}{g_3} \tag{4.17}$$

$$\dot{\tilde{\alpha}} = -\frac{k_1 f_3}{g_3} \tilde{\theta}_1 - k_1 z \tilde{\theta}_2 \tag{4.18}$$

Inserting ω_u into $\dot{V}_2$ then yields

$$\dot{V}_2 = -k_1 e_1^2 - k_2 e_2^2 + e_2 \dot{\tilde{\alpha}} - \tilde{\theta}_3 f_3 e_1 + \tilde{\theta}_4 \dot{p}_r e_1 \tag{4.19}$$

$$\dot{V}_2 = -k_1 e_1^2 - k_2 e_2^2 - \tilde{\theta}_1 \frac{e_2 k_1 f_3}{g_3} - \tilde{\theta}_2 e_2 k_1 z - \tilde{\theta}_3 f_3 e_1 + \tilde{\theta}_4 \dot{p}_r e_1 \tag{4.20}$$

$$\tag{4.21}$$

In order to obtain the update laws for $\dot{\hat{\theta}}_i$ where $i \in \{1,2,3,4\}$ The following Lyapunov candidate is chosen

$$V_3 = V_2 + \frac{1}{2\gamma_i} \tilde{\theta}_i, \qquad i \in [1,2,3,4] \tag{4.22}$$

$$\dot{V}_3 = -k_1 e_1^2 - k_2 e_2^2 + \tilde{\theta}_1 \left(\frac{1}{\gamma_1} \dot{\hat{\theta}}_1 - \frac{e_2 k_1 f_3}{g_3} \right) + \tilde{\theta}_2 \left(\frac{1}{\gamma_2} \dot{\hat{\theta}}_2 - e_2 k_1 z \right) + \tilde{\theta}_3 \left(\frac{1}{\gamma_3} \dot{\hat{\theta}}_3 - f_3 e_1 \right) + \tilde{\theta}_4 \left(\frac{1}{\gamma_4} \dot{\hat{\theta}}_4 + \dot{p}_r e_1 \right) \tag{4.23}$$

Choosing the following update laws

$$\dot{\hat{\theta}}_1 = \frac{\gamma_1 e_2 k_1 f_3}{g_3} \qquad \dot{\hat{\theta}}_2 = \gamma_2 e_2 k_1 z \qquad \dot{\hat{\theta}}_3 = \gamma_3 f_3 e_1 \qquad \dot{\hat{\theta}}_4 = -\gamma_4 \dot{p}_r e_1 \tag{4.24}$$

yields

$$\dot{V}_3 = -k_1 e_1^2 - k_2 e_2^2 \tag{4.25}$$

Asymptotic tracking is achieved and the adaptive controller with its update law for the second order system can be written as

$$\omega_u = -g e_1 + \dot{\hat{\alpha}} - k_2 e_2 \tag{4.26a}$$

$$\dot{\hat{\theta}}_1 = \frac{\gamma_1 e_2 k_1 f_3}{g_3} \tag{4.26b}$$

$$\dot{\hat{\theta}}_2 = \gamma_2 e_2 k_1 z \tag{4.26c}$$

$$\dot{\hat{\theta}}_3 = \gamma_3 f_3 e_1 \tag{4.26d}$$

$$\dot{\hat{\theta}}_4 = \gamma_4 \dot{p}_r e_1 \tag{4.26e}$$

4.1.1 Controller Validation 2nd Order Adaptive Backstepping Controller

Figure 4.1 displays the p_r step changes simulation with the 2nd order adaptive integrator backstepping controller and the 2nd order design model. During this simulation, the parameters in the controller are equal to the ones listen in table 4.1. It is clear that the controller manages to track the reference p_r with the choke pressure p_c. However, looking at the position z and the control output ω_u it is clear that the controller causes oscillations during the reference changes. The reason for this behavior is due to the term $-g_1 e_1$. This term behaves better in the adaptive controller compared to the non-adaptive controller. This is due to $\frac{\beta}{\sqrt{\rho}}$ being removed from the term, causing the magnitude of g_1 to be significantly smaller. However, the term still causes problems for the adaptive controller. Since the adaptive controller relies on the update laws $\hat{\theta}_1 - \hat{\theta}_4$ in order to prove stability, cascade theory can not be used in order to prove stability of the adaptive controller without the term $-g_1 e_1$, since it will not cover the stability of the update laws.

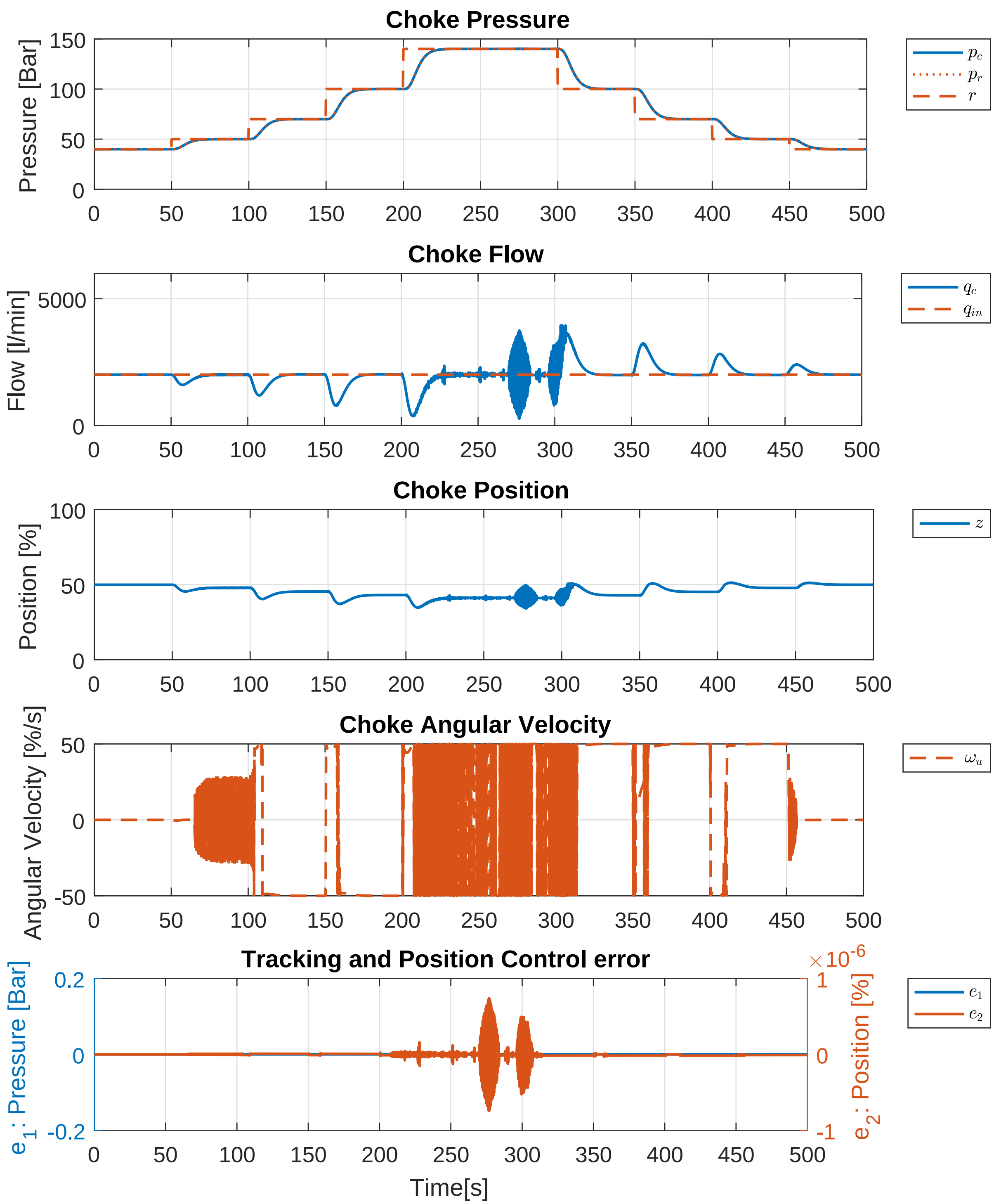

Figure 4.1: Plot results from a simulation of the 2nd order adaptive integrator backstepping controller with the 2nd order design model in the p_r step changes scenario

β	Bulk modulus	1×10^9 Pa
ρ	Density of fluid	$1500\,\mathrm{kg\,m^{-3}}$
V	Annulus volume	$150\,\mathrm{m^3}$
K_c	Choke gain	0.002850
k_1	Controller gain 1	1×10^{-7}
k_2	Controller gain 2	10
γ_1	Update law 1 gain	1×10^7
γ_2	Update law 2 gain	1×10^4
γ_3	Update law 3 gain	1
γ_4	Update law 4 gain	1×10^{-18}

Table 4.1: Parameters for 2nd order adaptive integrator backstepping controller during simulations with the 2nd order design model

Figure 4.2 shows a simulation with the 2nd order adaptive controller during the connection scenario. In this case the controller manages to keep the pressure p_c on the reference value p_r during both ramp down and ramp up, with the biggest offset being less that 5×10^{-6} bar. In this scenario, there are no oscillations caused by the controller, indicating that the controller is capable of handling flow changes in q_{in}. However, since the controller already behaved badly during the p_r step changes scenario, and considering that this simulation is on a perfect system, the performance of the adaptive 2nd order controller is not satisfactory.

Due to the behavior of the adaptive controller for the second order system, an adaptive controller for the 3rd order system will not be developed in this thesis. The same term ge_1 will appear in a 3rd order controller and likely cause the same problems.

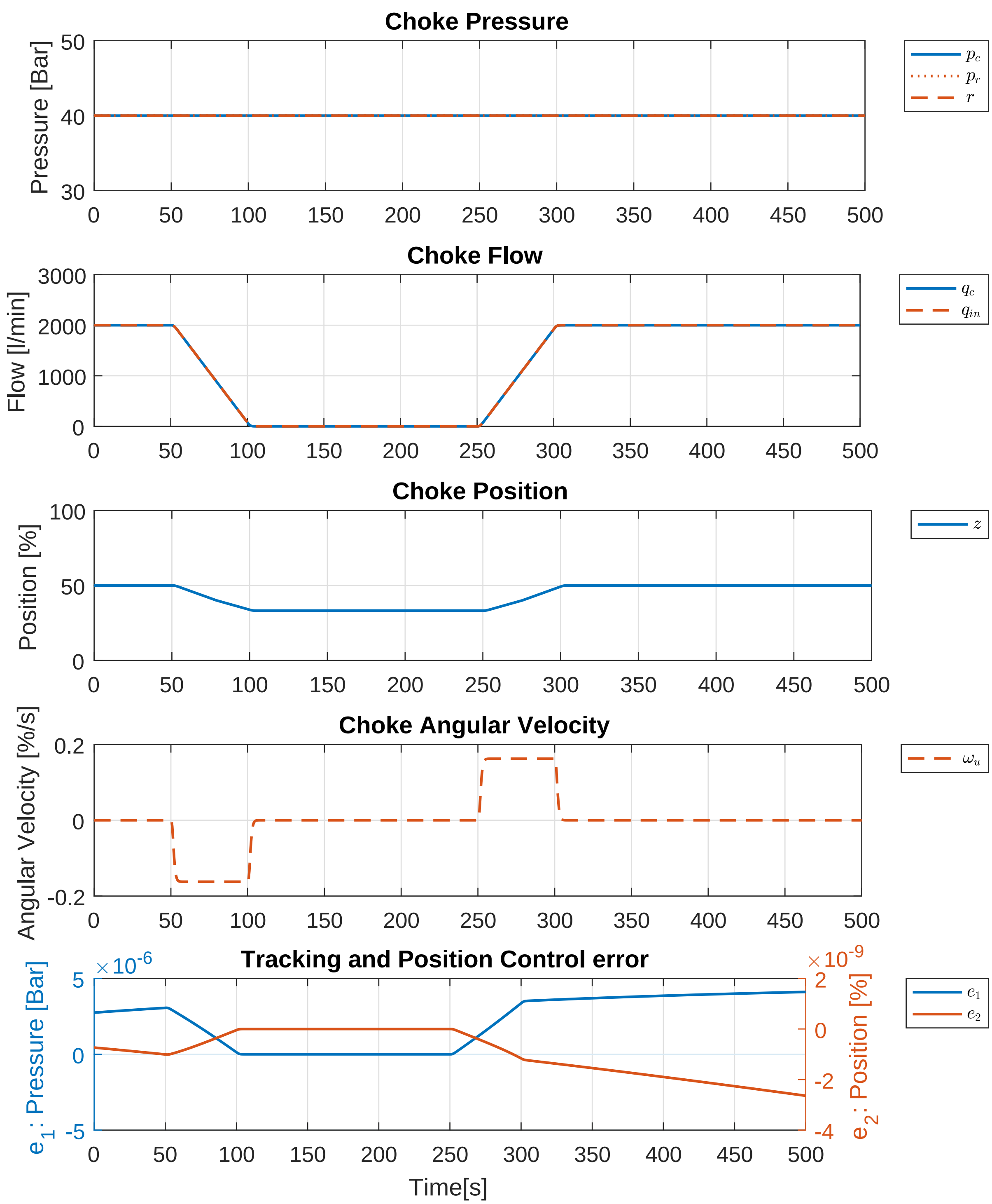

Figure 4.2: Plot results from a simulation of the 2nd order adaptive integrator backstepping controller with the 2nd order design model in the connection scenario

Chapter 5

Straume Validation Tests

This chapter covers validation of the performance of the 2nd and 3rd order integrator backstepping controller when used with Straume®. The 2nd order adaptive integrator backstepping controller will not be used due to the bad performance in Section 4.1.1.

The simulations will use the p_r step changes scenario and the connection scenario. In addition, simulations with these scenarios, but with wrong parametrization for bulk modulus or density will be performed in order to determine the controllers robustness in terms of bad parametrization.

The Straume® version provided by Kelda contains several inputs and outputs. For the validation tests the inputs z_c and q_p, and the outputs q_c, q_a and p_c will be used.

z_c is the choke position for the choke in Straume®. Since the choke input takes the position rather than the angular velocity, parts of the dynamic models used for the control development will be used in order to obtain the choke position z from the control output ω_u.

Since the simulator requires the position as input for the choke, it becomes easy to implement the necessary mapping function $h(z)$ from Section 2.1. $h(z)$ will simply be implemented between the dynamic model and Straume®. In the simulations in this chapter, the values of both z and $h(z)$ will be plotted. $h(z)$ will be the actual output from the controller that will be used as input into Straume®, while the value of z will be used as a variable within the controller. By using the mapping function $h(z)$ as input, the actual choke opening within Straume® will be similar to z. This means that in order to understand the behavior of how the choke position affects the pressure in Straume®, it is important to focus on z rather than $h(z)$.

q_p is the pump flow from the mud pumps in the system. q_{in}, which is the flow used in the dynamic models used for the control developments, is not possible to manipulate directly in Straume® (nor in a real life scenario). In order to obtain the ramp changes in q_{in}, which is used for the connection scenarios in the validation testing, q_p will be used instead. q_p and q_{in} are directly related, and the flow behavior through the drillstring and annulus will impact how q_{in} will behave during the ramp changes in q_p

q_c is the choke flow in Straume®. This output will be used to observe how the controllers affect the choke flow in the system and its behavior will provide useful information for the validation tests.

q_a is an array containing 100 separate flow values evenly spread throughout the length of the annulus. By extracting the last value the flow q_{in} is obtained, providing the controller with a necessary input. q_{in} will be used together with q_p in the validation tests in order to observe how the choke behaves during the different scenarios.

p_c is the choke pressure in the system and will be used as an input for the controllers. It will also provide crucial information during the validation tests as it is the variable that should track the reference signal p_r.

5.1 2nd Order Integrator Backstepping Controller

In this section the performance of the 2nd order integrator backstepping controller will be validated. During the validation tests the parameters of the controller are given in Table 5.1.

In order to provide Straume® with the correct choke input the design model in Equation 5.1

$$\dot{z} = \text{sat}(\omega_u) \tag{5.1}$$

The state z from this design model will be used in the mapping function $h(z)$ and the choke input in Straume® will be given as in Equation 5.2

$$z_c = h(z) \tag{5.2}$$

The simulations consisting of wrong density parametrization had little to no impact at all on

β	Bulk modulus	1×10^9 Pa
ρ	Density of fluid	$1500\,\mathrm{kg\,m^{-3}}$
V	Annulus volume	$100\,\mathrm{m}^3$
K_c	Choke gain	0.002850
k_1	Controller gain 1	1
k_2	Controller gain 2	1

Table 5.1: Parameters for 2nd order integrator backstepping controller during Straume® validation simulations

the controllers performance. Plot results from these simulations are therefore omitted in this chapter, but can be found in Appendix B.1

5.1.1 Validation Test With Reference Value Step Changes Scenario

Simulating Straume® with the 2nd order integrator backstepping controller in the p_r step changes scenario yields the results in Figure 5.1. For each step change, the choke pressure p_c tracks p_r quite well. However, the error e_1 is never 0 % in the time periods p_c is steady state, meaning that there is a slight offset between p_c and p_r. There is also a slight offset between the flow q_c and q_{in}. This is due the mud being compressed before the choke, causing a slight decrease in flow.

It is also worth mentioning the good behavior of z and ω_u in this simulation. Neither of them has any unnecessary oscillations in order to achieve the wanted results in this simulation

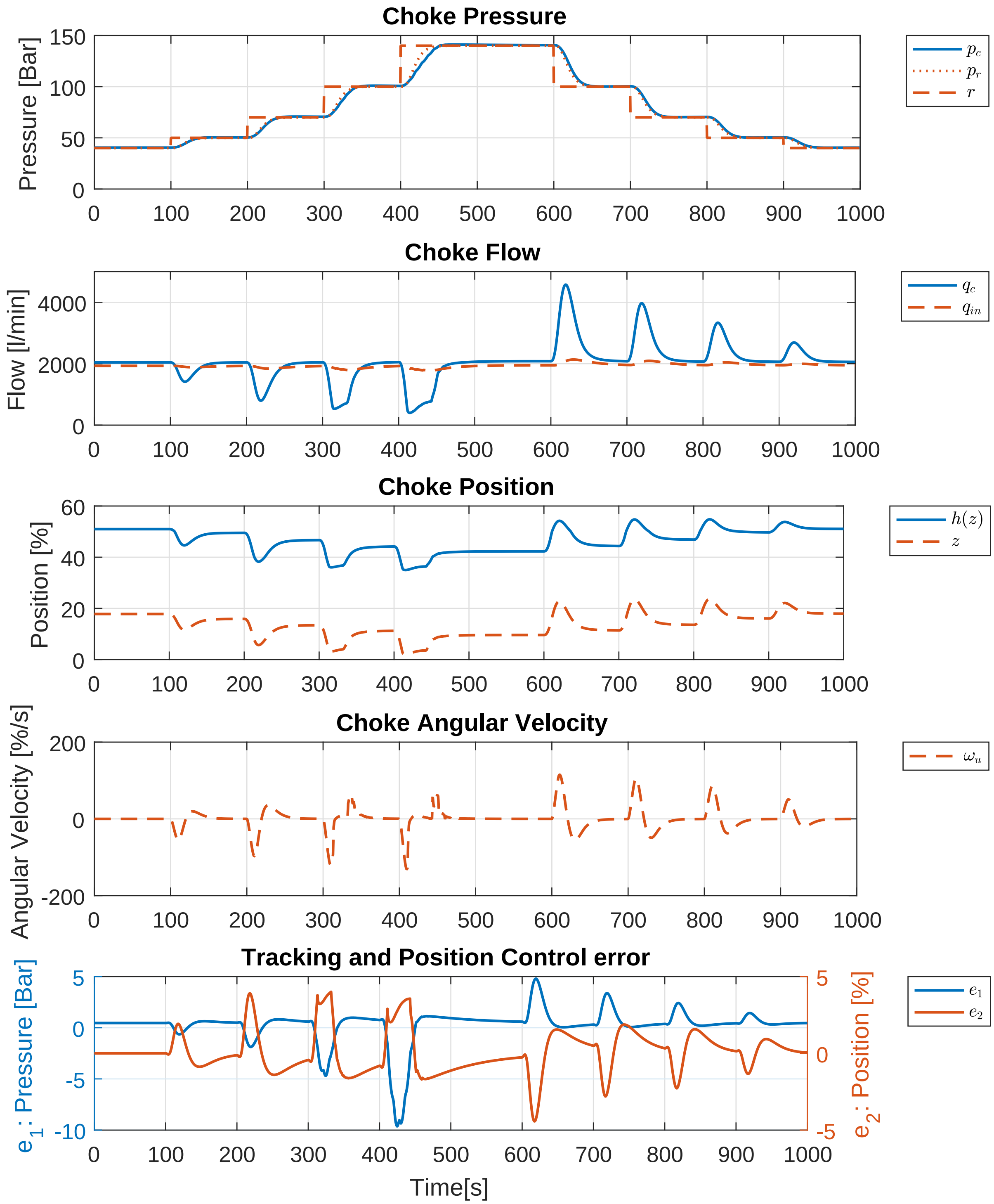

Figure 5.1: Plot results from a simulation of the 2nd order integrator backstepping controller with Straume® in the p_r step changes scenario

5.1.2 Validation Test with Connection Scenario

Figure 5.2 contains the plot results from simulating the Straume® simulator with the 2nd order integrator backstepping controller in the connection scenario. In this scenario, the controller manages to keep the pressure p_c close to the reference value p_r during the simulation, and during ramp down and ramp up the choke pressure also changes slightly. This is due to the ramping in q_{in} happening before the ramping in q_c. This can clearly be seen in the plot. During the ramp down, q_c slightly lags behind q_{in}, meaning that q_c is slightly larger than q_{in} in this period. This causes a small, yet constant decrease in pressure during the the ramp down. The opposite holds true during the ramp up. For a short period of time, q_{in} is larger than q_c, causing an increase in p_c for a short time during the ramp up. It's also important to note the offset between p_c and p_r during steady state. When the flow $q_{in} = 0$ it is impossible for the controller to do anything about the offset since it is necessary for the flow $q_{in} > 0$ (thus making it possible to achieve $q_{in} > q_c$) in order to produce an increase in the pressure p_c. However, there is also an offset between p_c and p_r before the ramp down and after the ramp up. In these periods the controller is presented with the necessary conditions in order to compensate for this offset. The reason for this offset not being dealt with is likely due to the lack of any term containing e_1 in the controller. Had the term $-ge_1$ still been a part of the controller, it might have canceled out the offset. An interesting situation in this scenario is the oscillations in the system happening just after the ramp down. This is likely due to the oscillations in q_{in} where the flow also holds negative values, meaning that the flow in the choke goes in the opposite direction.

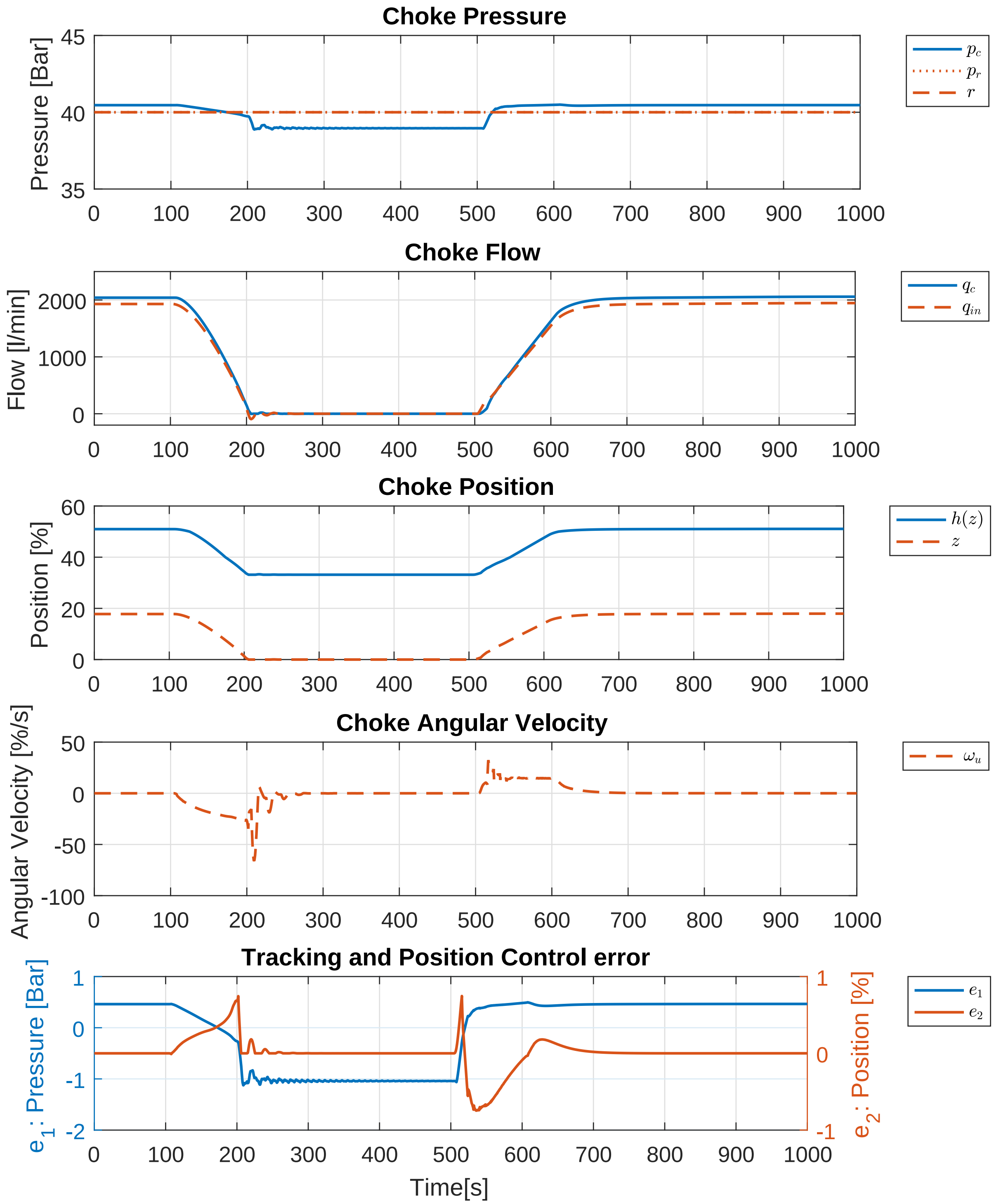

Figure 5.2: Plot results from a simulation of the 2nd order integrator backstepping controller with Straume® in the connection scenario

5.1.3 Validation Test with Reference Value Step Changes Scenario with Low Bulk Modulus

A simulation of the p_r step changes scenario is illustrated in Figure 5.3. In this simulation the value of the bulk modulus β within Straume® holds the value 1×10^8 Pa. This corresponds to 5 % gas in the fluid in the annulus. This is a high amount of gas, but still within what could be expected to happen during a drilling operation. It is clear from the simulation that this heavily affects the capability of tracking the reference signal p_r with the choke pressure p_c. However, looking at the behavior of ω_u and the choke position z it becomes clear that the controller is not to blame in this situation. In this simulation, the controller closes the choke almost completely, which is the correct choice in order to increase the pressure p_c. Due to the low bulk modulus β, the pressure p_c will simply not increase fast enough in order to keep up with the reference value p_r. In order to provide the controller with a validation test that properly validates the controllers capability of handling reference changes in p_r with a low bulk modulus β, the $p_r$steps validation test needs to be altered in order for the pressure p_c to actually be able to keep up with the reference value p_r.

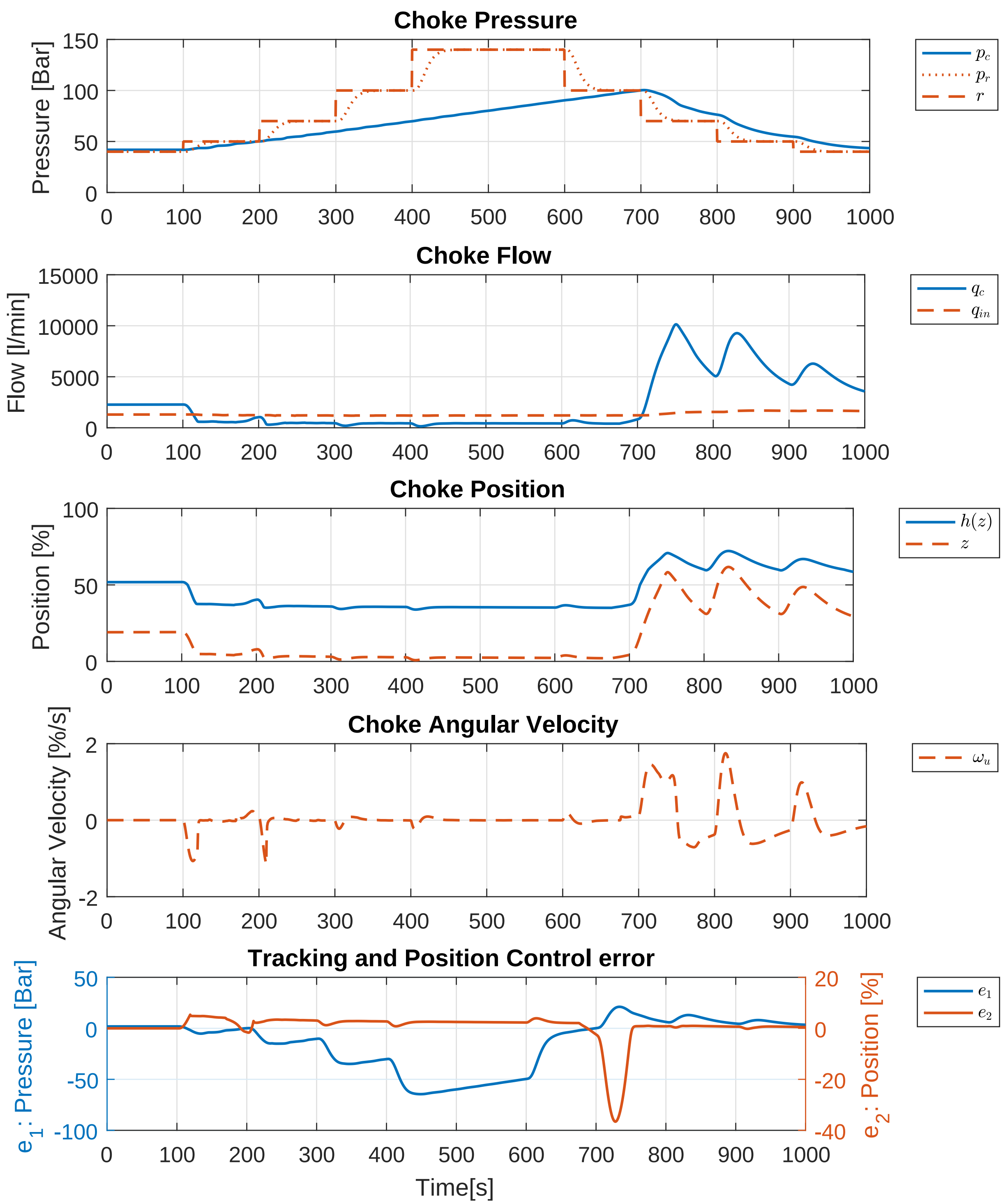

Figure 5.3: Plot results from a simulation of the 2nd order integrator backstepping controller with Straume® in the p_r step changes scenario with low bulk modulus in Straume

Figure 5.4 displays the results of simulating the 2nd order integrator backstepping controller against Straume® with an altered p_r step changes scenario. In this scenario there are fewer step changes, and the size of the steps are significantly smaller compared to the the simulation from Figure 5.3. In this simulation, the the pressure p_c is capable of following the reference changes in p_r, however, there is a significant offset between the two. One interesting point regarding this offset is that the offset remain the same in magnitude in between the step changes. This means that the controller does not manage to cancel out this offset over time, nor does the offset seem to increase. It is also noteworthy that both the offset between the choke pressure p_c and the reference value p_r and the offset between the flows q_{in} and q_c are much larger in this scenario compared to the same scenario that had correct parametrization for bulk modulus. This indicates that the flow difference could be responsible for the offset in pressure.

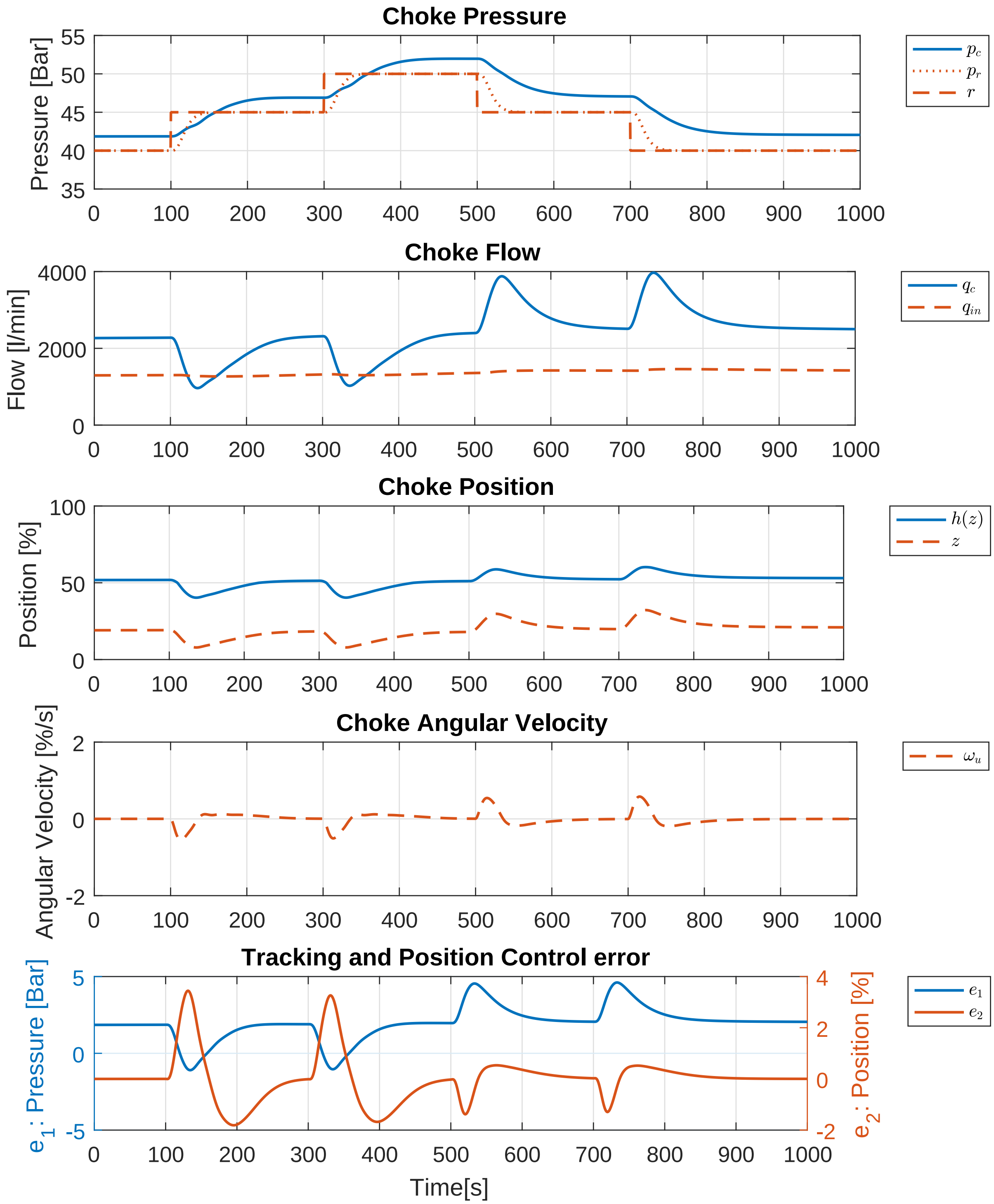

Figure 5.4: Plot results from a simulation of the 2nd order integrator backstepping controller with Straume® in the altered p_r step changes scenario with low bulk modulus in Straume

5.1.4 Validation Test With Connection Scenario with Low Bulk Modulus

A simulation of the connection scenario with the 2nd order integrator backstepping controller is illustrated in Figure 5.5. In this simulation the bulk modulus β in Straume® holds the value of 1×10^8 Pa. As with the simulation of the p_r step changes with low bulk modulus β in Section 5.1.3, there is an significant offset between the reference p_r and the choke pressure p_c. Compared to the same connection scenario from Section 5.2.2, the offset in this simulation is significantly bigger, indicating that the lower β value impacts this offset. In addition, this offset decreases as the flow q_{in} ramps down and when the flow $q_{in} = 0$ the offset is 0 as well. This indicates that the offset is largely due to the flow q_{in}. Apart from the offset, the controller seems to handle this scenario quite well, even with the β value being so far off.

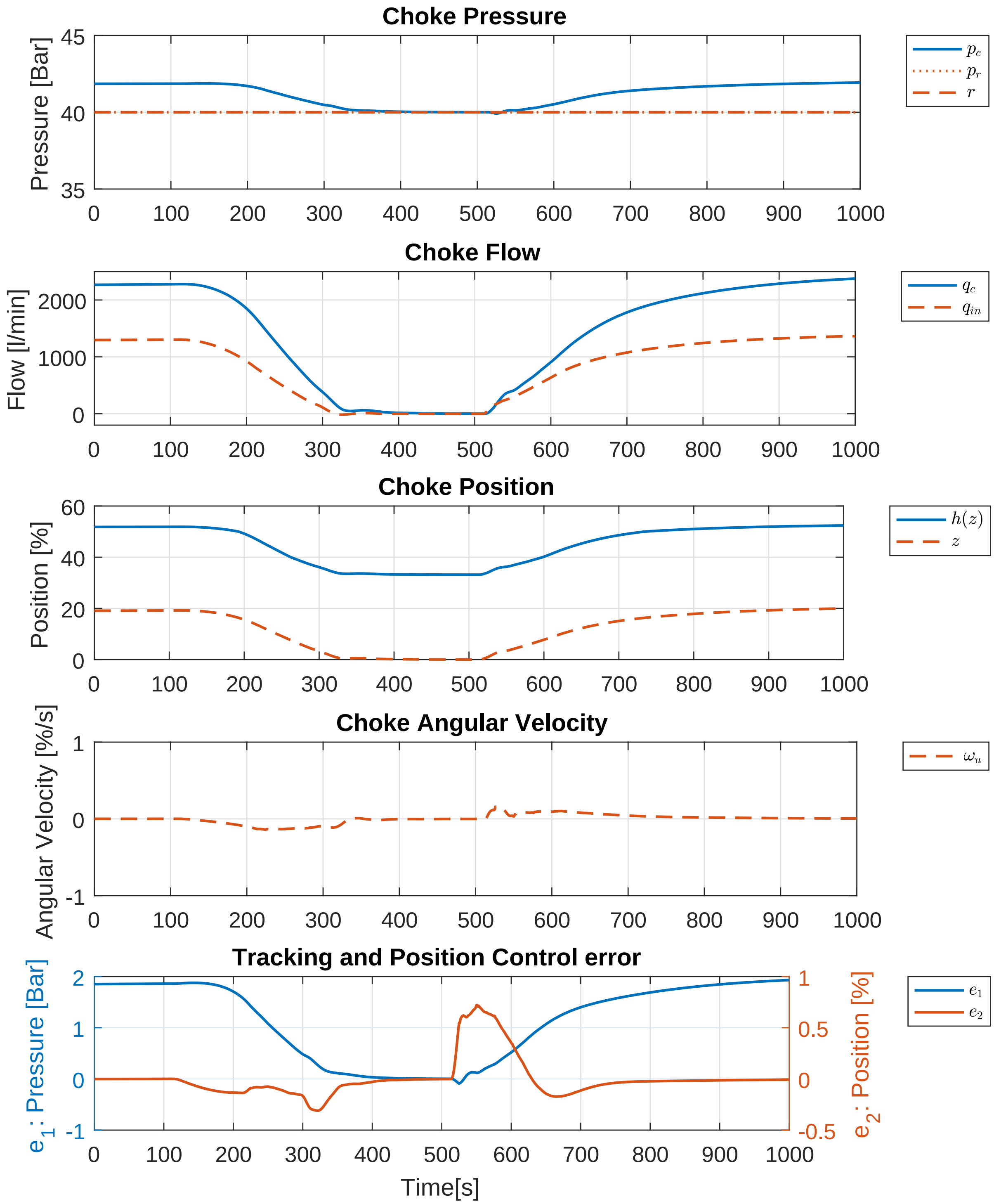

Figure 5.5: Plot results from a simulation of the 2nd order integrator backstepping controller with Straume® in the connection scenario with low bulk modulus in Straume

β	Bulk modulus	1.25×10^9 Pa
ρ	Density of fluid	$1500\,\mathrm{kg\,m^{-3}}$
V	Annulus volume	$100\,\mathrm{m}^3$
K_c	Choke gain	0.002850
k_1	Controller gain 1	1
k_2	Controller gain 2	1
k_3	Controller gain 3	1
τ_ω	actuator time constant	0.5

Table 5.2: Parameters for 3rd order integrator backstepping controller during Straume® validation simulations

5.2 3rd Order Integrator Backstepping Controller

In this section the performance of the 3rd order integrator backstepping controller will be validated in simulations with Straume®. During these simulations the controller's parameters are given in Table 5.2.

In order to provide Straume® with the correct input, the the dynamic model in Equation (5.3b) is implemented.

$$\dot{z} = \omega \tag{5.3a}$$

$$\dot{\omega} = \frac{1}{\tau_\omega}\big(-\omega + \mathrm{sat}(\omega_u)\big) \tag{5.3b}$$

From this dynamic model, the state z can be used as an input for z_c in Straume®. However, in order to compensate for the mapping of the position z_c to the actual area choke opening in the choke within Straume®, the mapping function $h(z)$ is added to the system, i.e.

$$z_c = h(z) \tag{5.4}$$

The simulations consisting of wrong density parametrization had little to no impact at all on the controllers performance. Plot results from these simulations are therefore omitted in this chapter, but can be found in Appendix B.2

5.2.1 Validation Test with Reference Value Step Changes Scenario

Simulating Straume® with the p_r step changes scenario with the 3rd order integrator backstepping controller yields the results in Figure 5.6. It is clear that p_c tracks p_r quite well, however, there is a slight offset between the two during steady state. This is likely due to the lack of a term containing e_1 in the controller. It is possible that if $-ge_1$ was still a part of the controller and not ill-conditioned, the offset could have been compensated for. The controller also behaves well in this scenario as it does not produce any unnecessary oscillations in order to achieve the tracking.

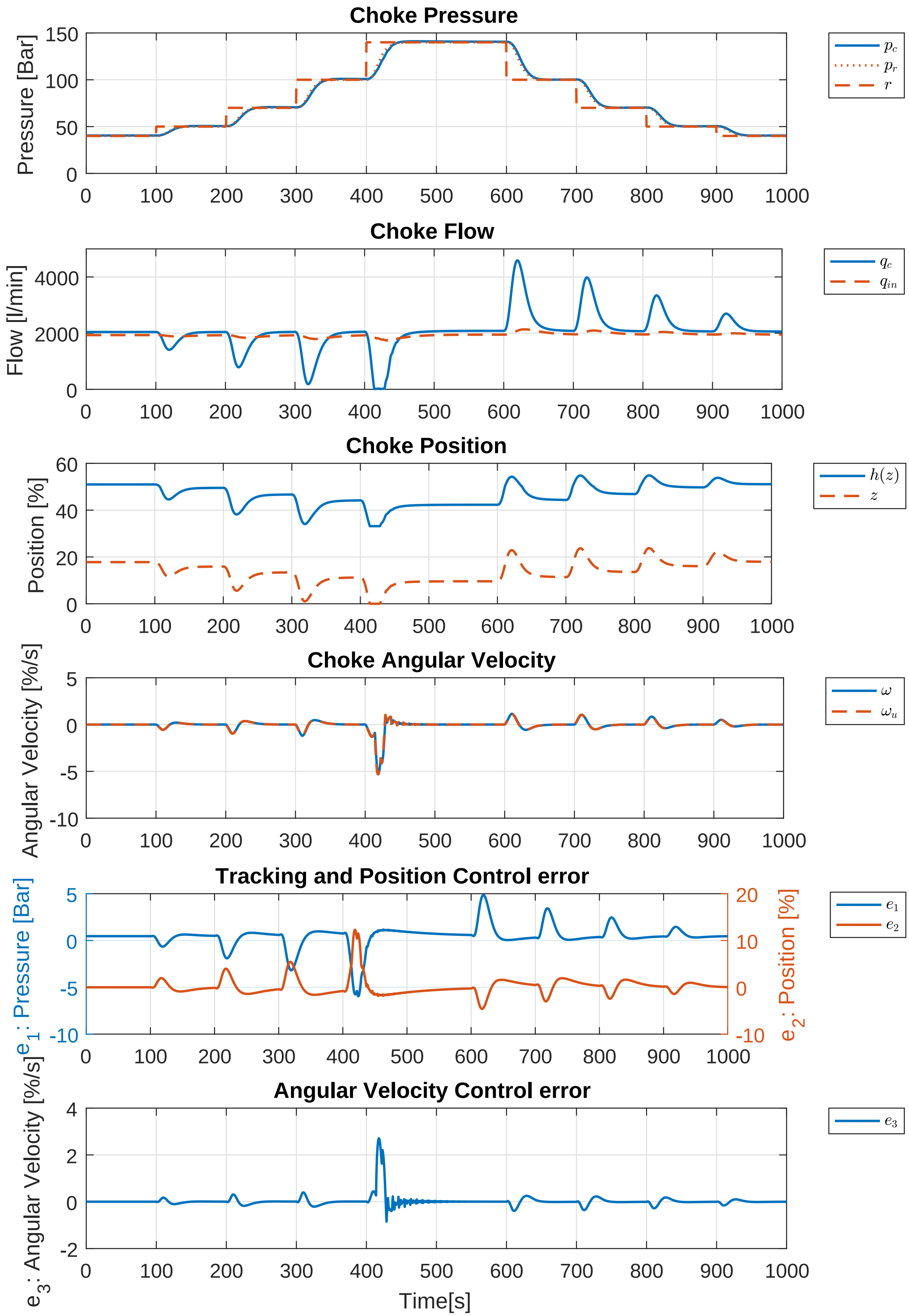

Figure 5.6: Plot results from a simulation of the 3rd order integrator backstepping controller with Straume® in the p_r step changes scenario

5.2.2 Validation Test with Connection Scenario

In Figure 5.7 the results from simulating Straume® in the connection scenario are plotted. The controller manages to keep p_c close to the reference value p_r, however, the offset is still quite noticeable. The results in this case are much alike the results from the same scenario, but with the 2nd order controller in terms of tracking, and the offset occurring this scenario is due to the same reasons as with the 2nd order controller. Unlike the 2nd order controller, the 3rd order controller output ω_u produces quite a lot of oscillations. These oscillations are especially bad when the flow q_{in} reaches 0, but there are also oscillations during the ramp up of the flow q_{in}. This indicates that the 3rd order controller does not only behave poorly during low flow, but it seems to handle flow changes bad in general.

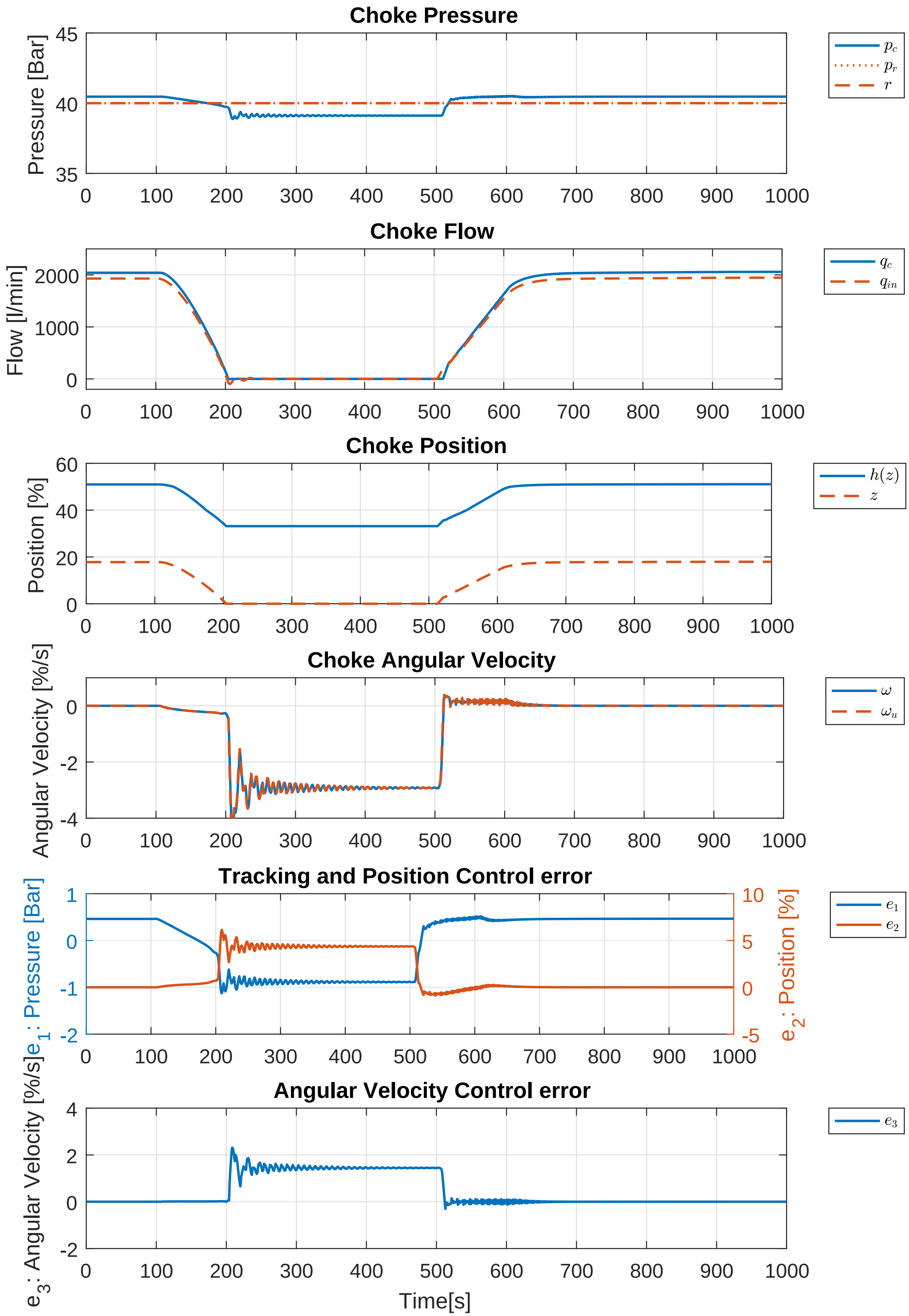

Figure 5.7: Plot results from a simulation of the 3rd order integrator backstepping controller with Straume® in the connection scenario

5.2.3 Validation Test with Reference Value Step Changes Scenario with Low Bulk Modulus

A simulation of the altered p_r step changes scenario (from Section 5.1.3) with a bulk modulus value of $\beta = 1 \times 10^8$ Pa in Straume® is performed with the 3rd order integrator backstepping controller in Figure 5.8. In this simulation the controller is capable of tracking the reference value p_r with the choke pressure p_c quite well. However, there is a noteworthy offset between the reference value p_r and the choke pressure p_c in this simulation. This offset does not change in magnitude between the step changes, meaning that the controller does not manage to decrease this offset during the step changes.

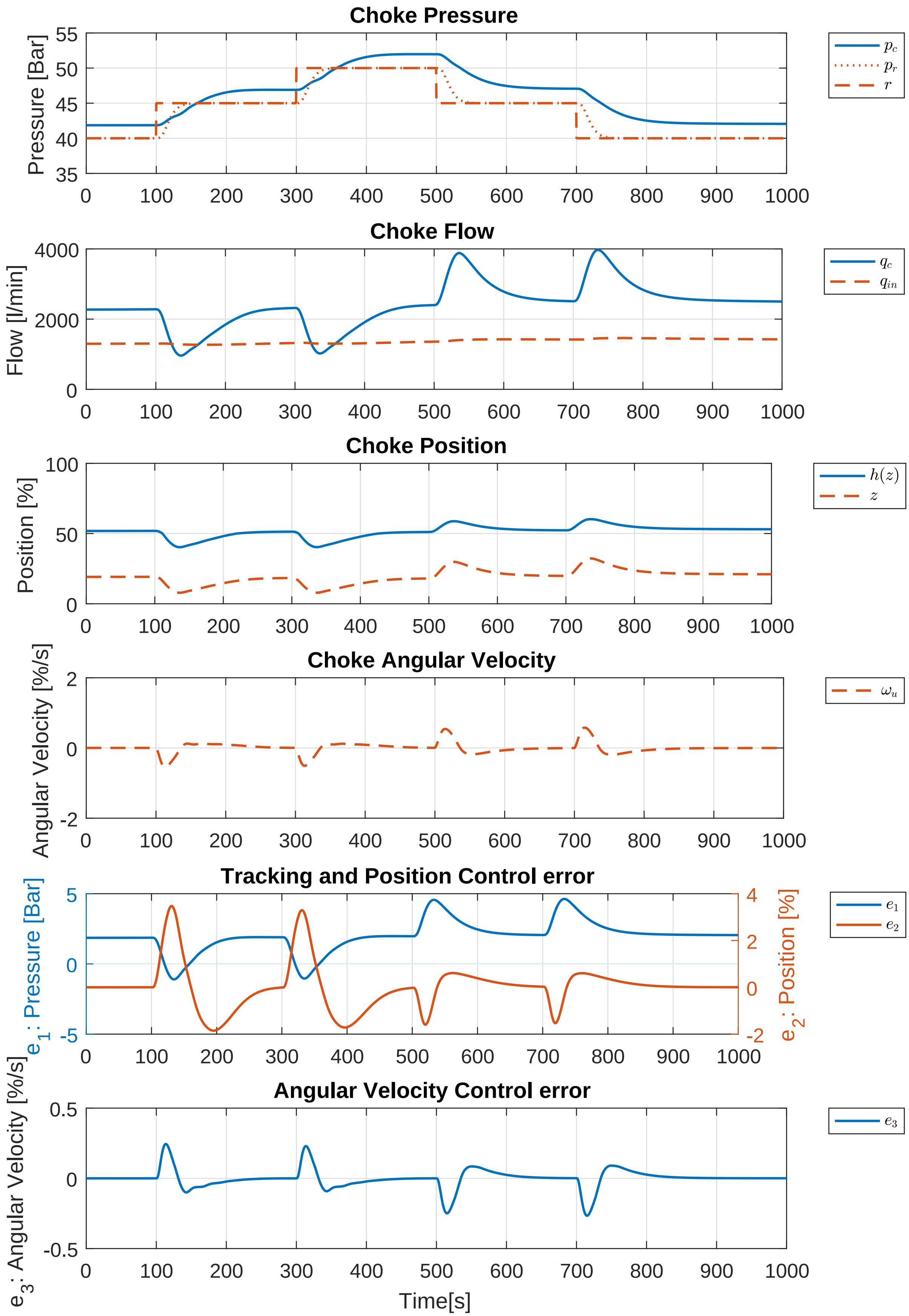

Figure 5.8: Plot results from a simulation of the 3rd order integrator backstepping controller with Straume® in the p_r step changes scenario with low bulk modulus

5.2.4 Validation Test with Connection Scenario with Low Bulk Modulus

Figure 5.9 displays the results from simulating the 3rd order integrator backstepping controller with Straume in the connection scenario with the bulk modulus in Straume® being 1×10^8 Pa. The controller manages to some degree to maintain the pressure during ramp down and ramp up in q_{in}, however, there is an offset before the ramp down and after the ramp up. This indicates that the offset caused is largely due to q_{in}, since the offset disappears when $q_{in} = 0$. Since the offset before and after the connection in this simulation is larger than the offset in the same simulation, but with the correct value for β in Section 5.2.2, indicates that the bulk modulus β also impacts this offset.

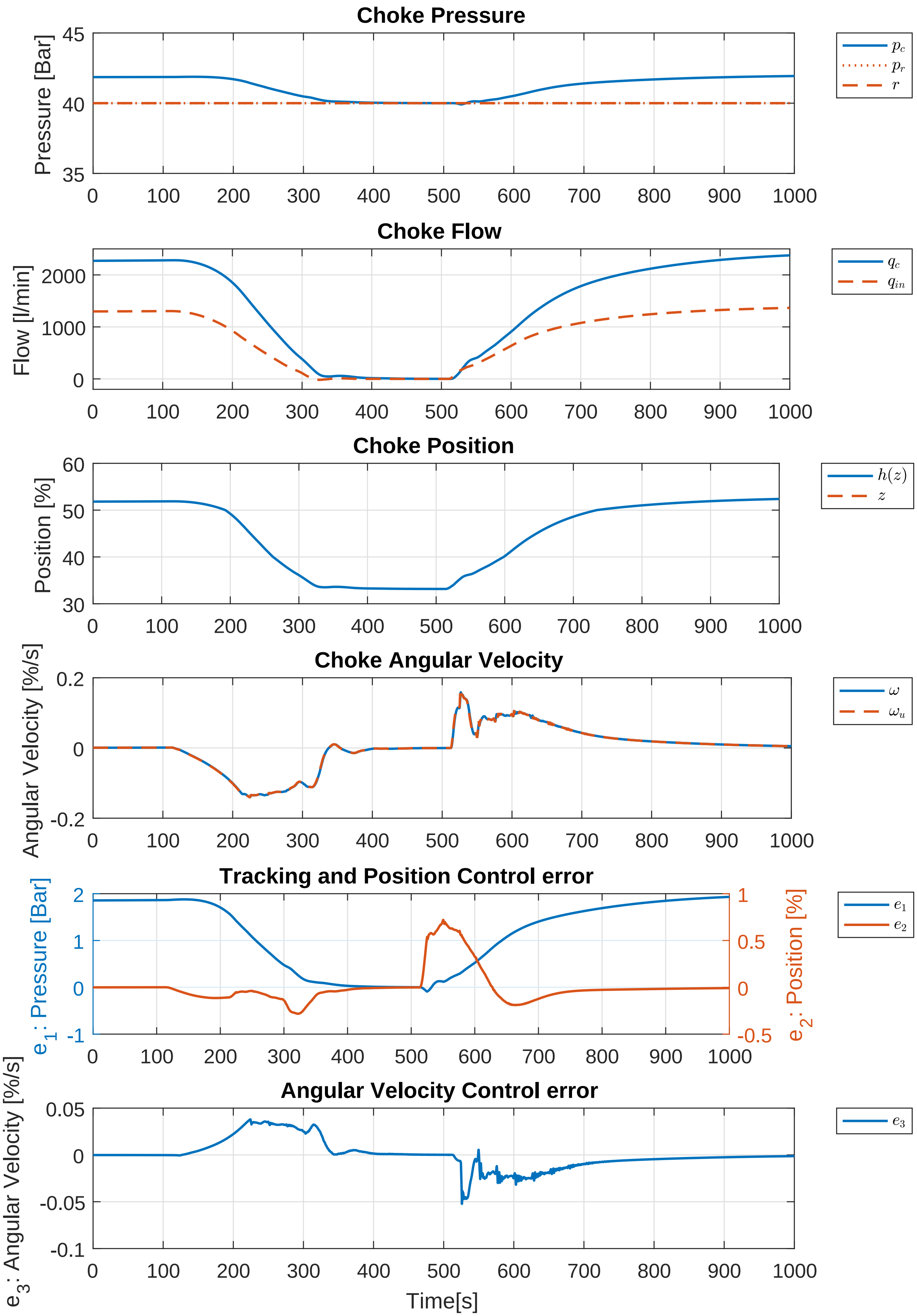

Figure 5.9: Plot results from a simulation of the 3rd order integrator backstepping controller with Straume® in the connection scenario

Chapter 6

Summary and Recommendations for Further Work

This chapter covers the results from this project and a discussion around the performance of the developed controllers. It also cover recommendations for further work in order to improve the performance of the controllers developed in this thesis.

6.1 Summary and Conclusions

In MPD one relies on good choke control in order to obtain the correct pressure at the bottom of the well during drilling operations. In this thesis, controllers based on integrator backstepping and adaptive integrator backstepping were developed in order to obtain an understanding on how well these controllers manage to control the choke pressure. The controllers should be able to control the choke pressure such that it follows a reference value controlled by an operator, and should be able to achieve this by adjusting the choke's angular velocity.

The controllers were developed based on dynamic models for the choke pressure and then simulated in two different scenarios that validated the controllers performance. The controllers would be simulated with the dynamic models that was used in their development, and thus the conditions during the simulation is considered perfect for the controller. The first scenario consisted of a series of increasing and decreasing step changes in the pressure reference value. In this scenario the controller should be able to change the choke pressure in order to track the reference value. The second scenario simulated a connection, a procedure were the flow into the choke decreases until there is no flow and then increases the flow until it reaches its original value. In this scenario the controller should be able to maintain the choke pressure during flow changes. These tests were designed to verify whether the controller can handle normal operations during drilling, and since the conditions are perfect, these tests verify if the controllers simply works as expected.

Good performing controllers would then go through further testing with Straume®, a high-end multiphase well simulator. When simulated with Straume®, the controllers would go through the reference value changes scenario and the connection scenario in order to verify whether the controllers handles normal operations on a system that give a better representation of oil-drilling in reality. In addition, the controllers would go through the same scenarios, but with bad parameterization for the bulk modulus and density. This would test the controllers robustness in terms of handling wrong parameterization of the often unknown parameters.

Chapter 2 covers the development of the dynamic models used to describe the change of choke pressure in the system. The first dynamic model was of 2nd order. This model described how the choke pressure changed based on the flow in and out of the choke were the flow out of the choke depended on the choke opening. The choke opening changed according to the angular velocity of the choke. The second model was of 3rd order and was an extension of the 2nd order model. The 3rd order model also featured the actuator dynamics in the choke which was given as a first order linear filter. Two dynamic models were used since the complexity of backstepping controllers drastically increases as the order of the system increases. A controller would first be developed for the 2nd order system, if it provided good results during testing, a new controller would be developed for the 3rd order system. This chapter also covers the development of a reference filter which would provide the backstepping controllers with a filtered reference value and its derivatives for the controller to track.

In Chapter 3 regular integrator backstepping controllers were developed. The first controller for the 2nd order model behaved badly due to an ill-conditioned term. Removing this term from the controller provided good results, however, this term appeared as part of the development in order to satisfy a Lyapunov function, and by removing it the proof of stability would no longer hold. The controller was instead proved stable by using cascade theory, and thus provided a well behaved controller for the 2nd order system. The controller for the 3rd order system encountered the same problem with the same ill-conditioned term. The ill-conditioned term was removed from the controller and cascade theory was used in order to prove that the controller for the 3rd order system was stable. The 3rd order controller also provided good results during testing.

An adaptive integrator backstepping controller was developed for the 2nd order system in Chapter 4. As with the regular integrator backstepping controllers, the adaptive controller performed badly during testing due to an ill-conditioned term. Unlike the regular controllers, the adaptive controller could not be proved stable using cascade theory. This is due to the adaptive controller relying on update laws for its parameter estimation and cascade theory does not provide stability proof for these update laws. Since the controller for the 2nd order system behaved badly, a controller for the 3rd order system was not developed. This is due to the ill-conditioned term that would also be a part of the 3rd order controller.

The regular integrator backstepping controllers based on the 2nd and 3rd order system performed well enough during the first tests. These controllers were simulated with Straume® in Chapter 5 and went through a series of validation tests. Since Straume® takes the choke position as input rather than the choke's angular velocity, the 2nd and 3rd order design models were used in order to provide Straume® with the choke position based on the controllers angular velocity output. This is not optimal since the actuator dynamics (which is the main difference between the controllers based on the 2nd and 3rd order model) is modeled outside of Straume®, and the advantage the 3rd order model could have because of this disappears due to this setup.

In the validation tests, the 2nd order controller handled the changes in reference value scenario well, even though there was a small offset between the reference value and the choke pressure. During the connection scenario, the 2nd order controller managed to maintain the choke pressure quite well, however, it was a notable offset between the choke pressure and the reference value in this scenario as well. These offset could might have been handled properly by the controller if the ill-conditioned term was still a part of the controller (and not ill-conditioned). Running the same scenarios with wrong parameter value for density had little to no impact on the performance of the controller. Running the same scenarios with wrong parameter values for bulk modulus (lower bulk modulus in Straume® than in the controller), however, had a significant impact on the performance. The changes in choke pressure became significantly slower, making it impossible for the controller to actually keep up with the reference value. In addition, the offset between the the reference value and choke pressure became significantly larger compared to the scenarios with correct bulk modulus.

The backstepping controller based on the 3rd order performed much like the controller based on the 2nd order system during all scenarios with a few exceptions. The 3rd order controller had some small yet noteworthy oscillations in the control output during the connection scenario. These oscillations could also be seen in the choke pressure. The control output also produced some very small oscillations during the step changes scenario, however, in this simulation the oscillations could not be seen in the choke pressure.

6.2 Discussion

In this thesis, three controllers were developed based on integrator backstepping. Two regular integrator backstepping controllers developed based on a 2nd and 3rd order design model, and an adaptive integrator backstepping controller based on a 2nd order dynamic model. The adaptive controller behaved badly even in the perfect case simulations and were not considered for further testing. The regular integrator backstepping controllers performed well when simulated with Straume during normal operation with the exception of a small offset between the reference value and the choke pressure. Also, the 3rd order controller had some oscillations in its output. These oscillations had notable impact during the connection scenario where the choke pressure started oscillating. Both controllers performed significantly worse when the bulk modulus parameter was wrong and both controllers had an increase in offset between the choke pressure and its reference value during these simulations.

Based on these finding both the regular integrator backstepping controllers can be used in order to control the choke, but in doing so one will need to be aware of the constant offset between the reference value and the choke pressure. It is also important to be aware of the significant impact wrong parametrization of the bulk modulus has on the controllers capability of tracking the reference signal. Even if it is possible to use the controllers, it is recommended that the current issues are dealt with before the controllers are used.

6.3 Recommendations for Further Work

The term $g_1 e_1$ caused a lot of problems in the controller development throughout this thesis. The term was ill-conditioned and caused all controllers to behave badly when developed based on integrator backstepping theory. Because of this a lot of time was spent on proving stability for the regular integrator backstepping controllers using cascade theory and a working adaptive integrator backstepping controller was not developed in the end.

For further work it is recommended to find a better way to deal with this ill-conditioned term. This could be solved by using a different Lyapunov function during the development of the integrator backstepping controllers. This Lyapunov function would either need to provide a controller without this ill-conditioned term or somehow manage to compensate for the large oscillations the ill-conditioned term creates.

Another solution in order to deal with the ill-conditioned term could be to change the units within the controller in such a way that the ill-conditioned terms magnitude becomes significantly smaller. If this ill-conditioned term is dealt with properly, the performance of the regular integrator backstepping controllers might increase significantly and the adaptive integrator backstepping controller might actually work properly.

A Lyapunov approach that handles the ill-condition term properly will also have to deal with the saturations of the choke position and angular velocity properly in the Lyapunov stability proof.

During the validation tests with Straume®, the 2nd and 3rd order design models were used in order to provide Straume® with the choke position based on the controllers angular velocity output. This means that the actuator dynamics that the 3rd order controller is designed to handle is not properly tested during these validation tests. It is recommended that Straume® gets an update were the simulator consists of well modeled choke dynamics that takes the angular velocity as input. If this is done and the same validation tests are performed without the use of design models, but with the controller output directly connected to Straume®, there is a chance that the differences between the 2nd and 3rd order controller become more apparent.

In order to perform the backstepping procedure, a mapping function $h(z)$ was added in the dynamic models during controller development in order to compensate for the mapping function $g(z)$ that naturally occurs in the choke. The addition of this mapping function came with two important criteria. The first is that the mapping function $g(z)$ is known and the second is that there must be possible to actually add the mapping function $h(z)$ before the choke position input. These criteria might be hard to accomplish and for further work it is recommended to find a way to develop integrator backstepping controllers without the use of the mapping function $h(z)$. This could be accomplished by using different virtual states or use a Lyapunov function that handles this mapping function properly during the controller development.

Appendix A

Acronyms

MPD Managed Pressure Drilling

UGES Uniform Exponential Asymptotic Stability

UGES Uniform Global Asymptotic Stability

UGB Uniformly Globally Bounded

Appendix B

Straume Simulations with Wrong Density

This appendix holds the results from the simulations of the 2nd and 3rd order integrator backstepping controllers during simulations with Straume with wrong parametrization for density. The density in the controllers are $\rho = 1500\,\mathrm{kg\,m^{-3}}$. The density in Straume® is $\rho = 900\,\mathrm{kg\,m^{-3}}$ during the simulations with low density and $\rho = 1800\,\mathrm{kg\,m^{-3}}$ during simulations with high density.

Information regarding the scenarios are found in the caption of each figure.

B.1 2nd Order Integrator Backstepping Controller

B.1.1 Reference value step changes scenario with low density

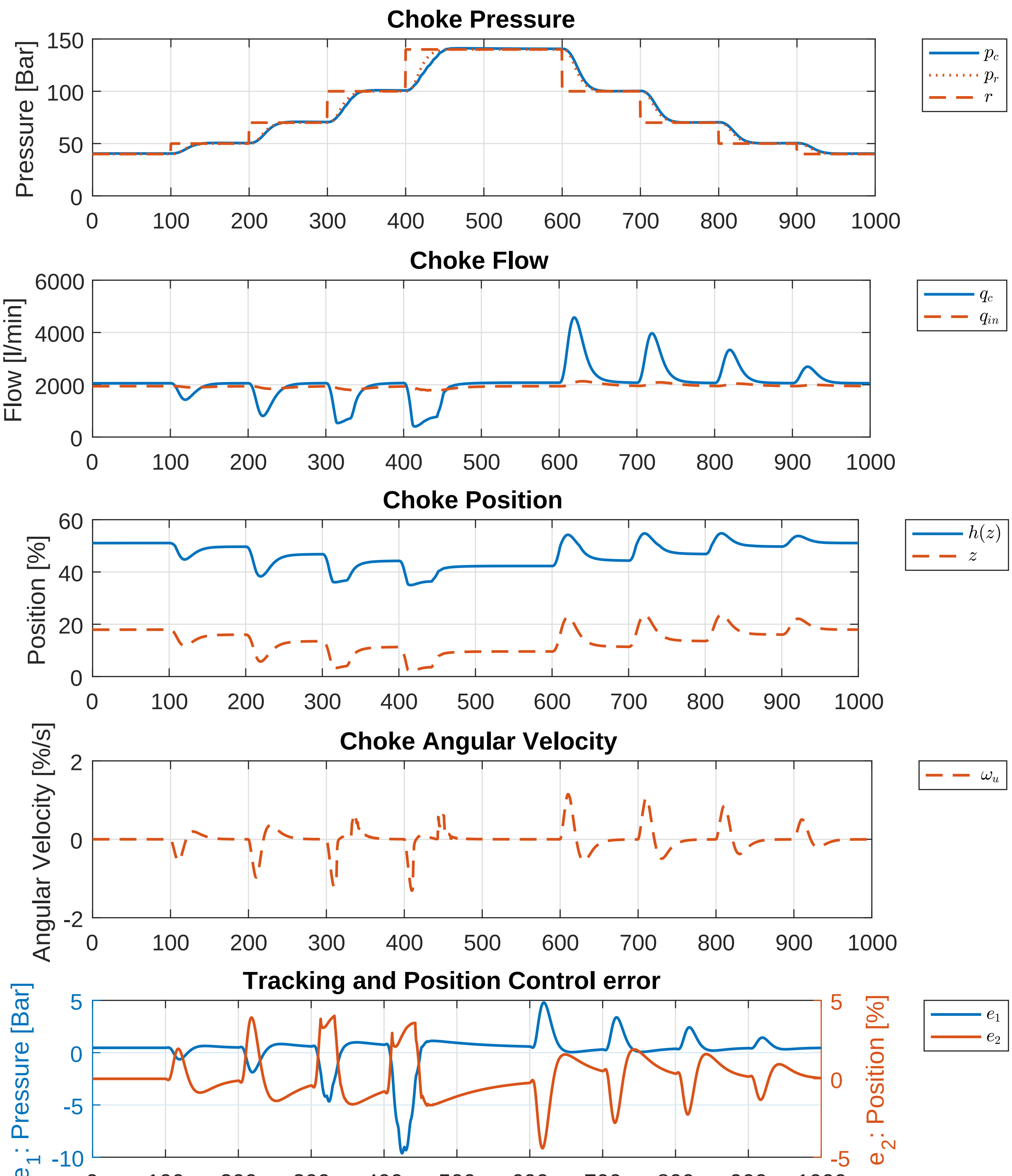

Figure B.1: Plot results from a simulation of the 2nd order integrator backstepping controller with Straume® in the p_r step changes scenario with low density in Straume

B.1.2 Reference value step changes scenario with high density

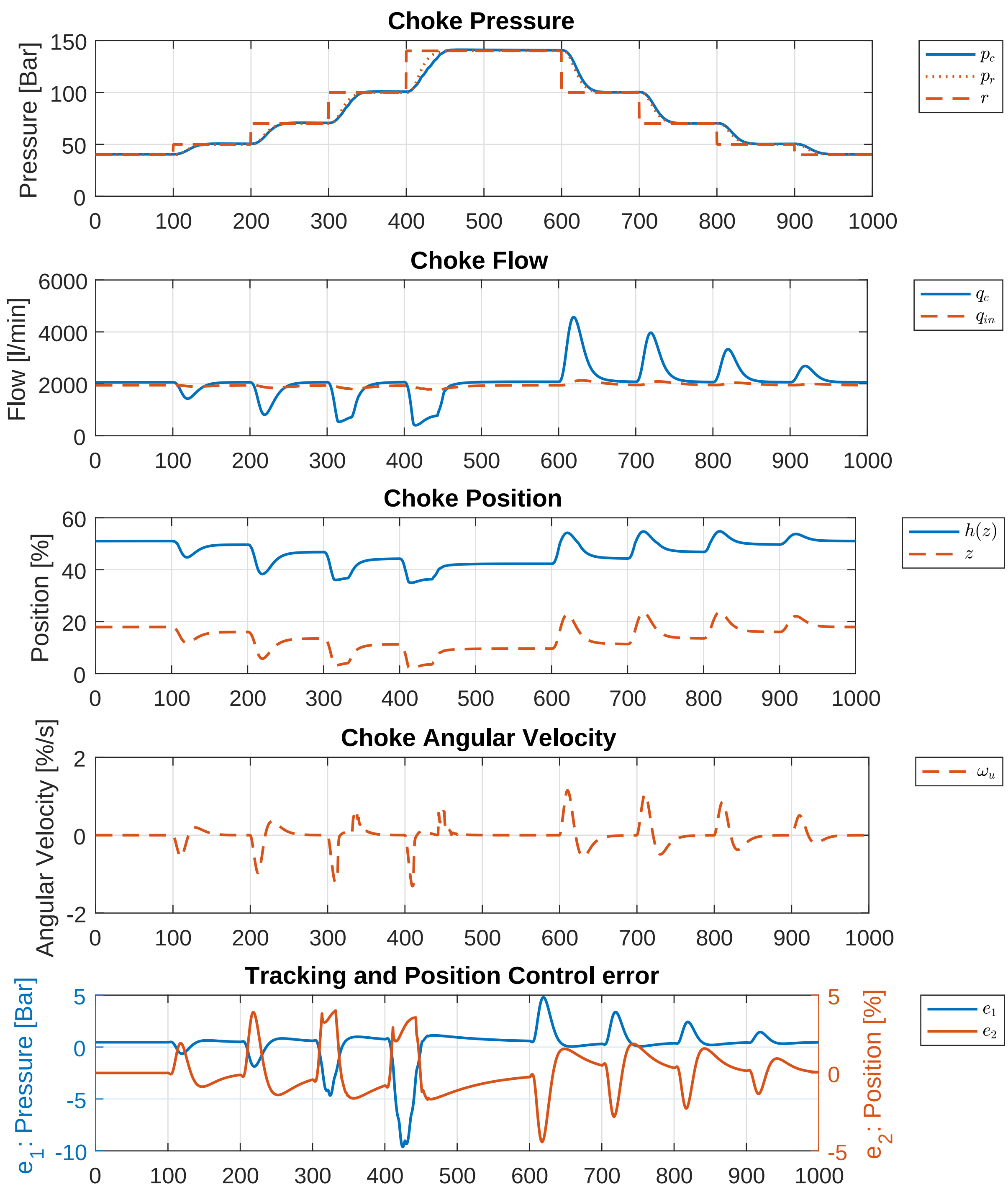

Figure B.2: Plot results from a simulation of the 2nd order integrator backstepping controller with Straume® in the p_r step changes scenario with high density in Straume

B.1.3 Connection scenario with low density

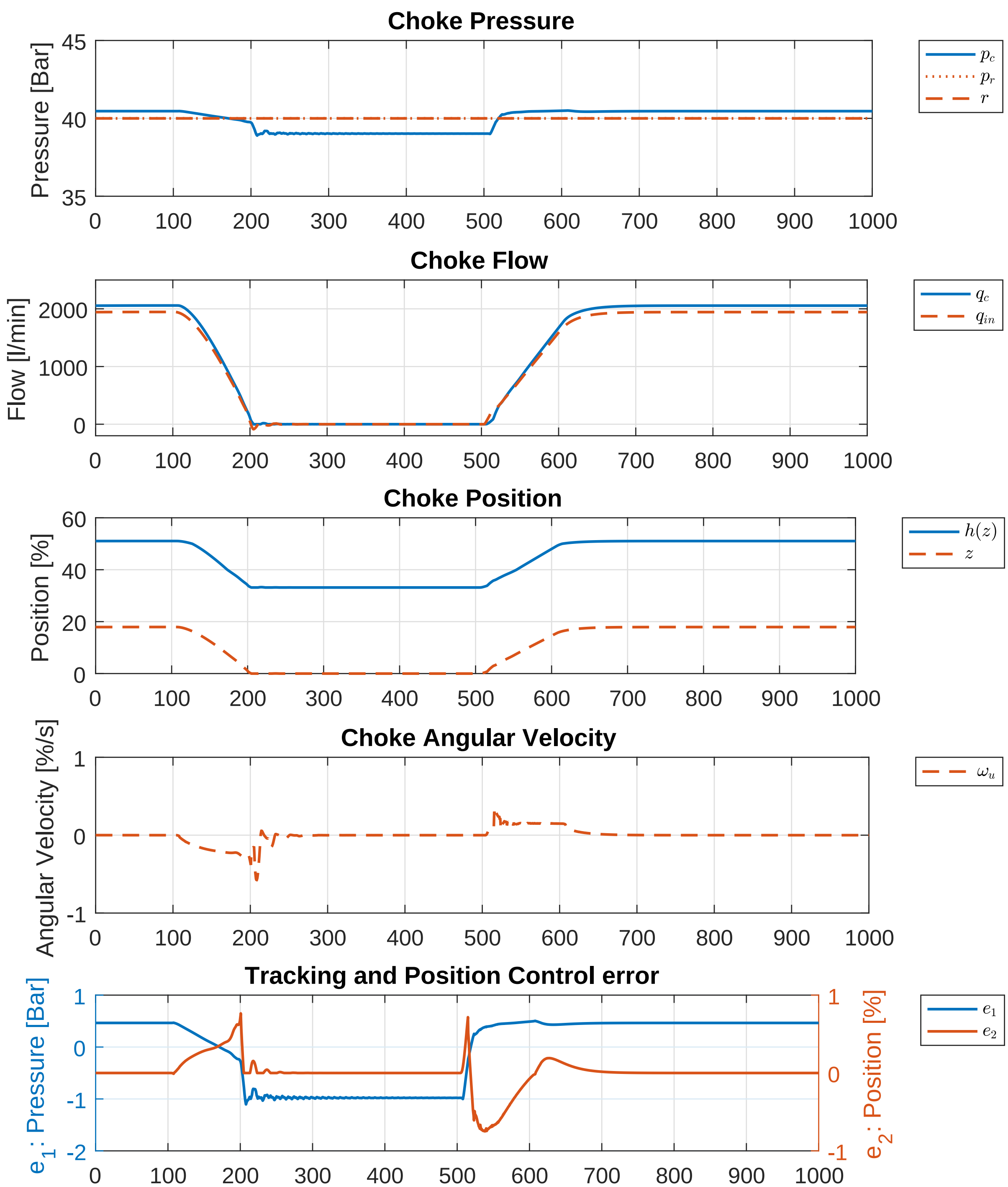

Figure B.3: Plot results from a simulation of the 2nd order integrator backstepping controller with Straume® in the connection scenario with low density in Straume

B.1.4 Connection scenario with high density

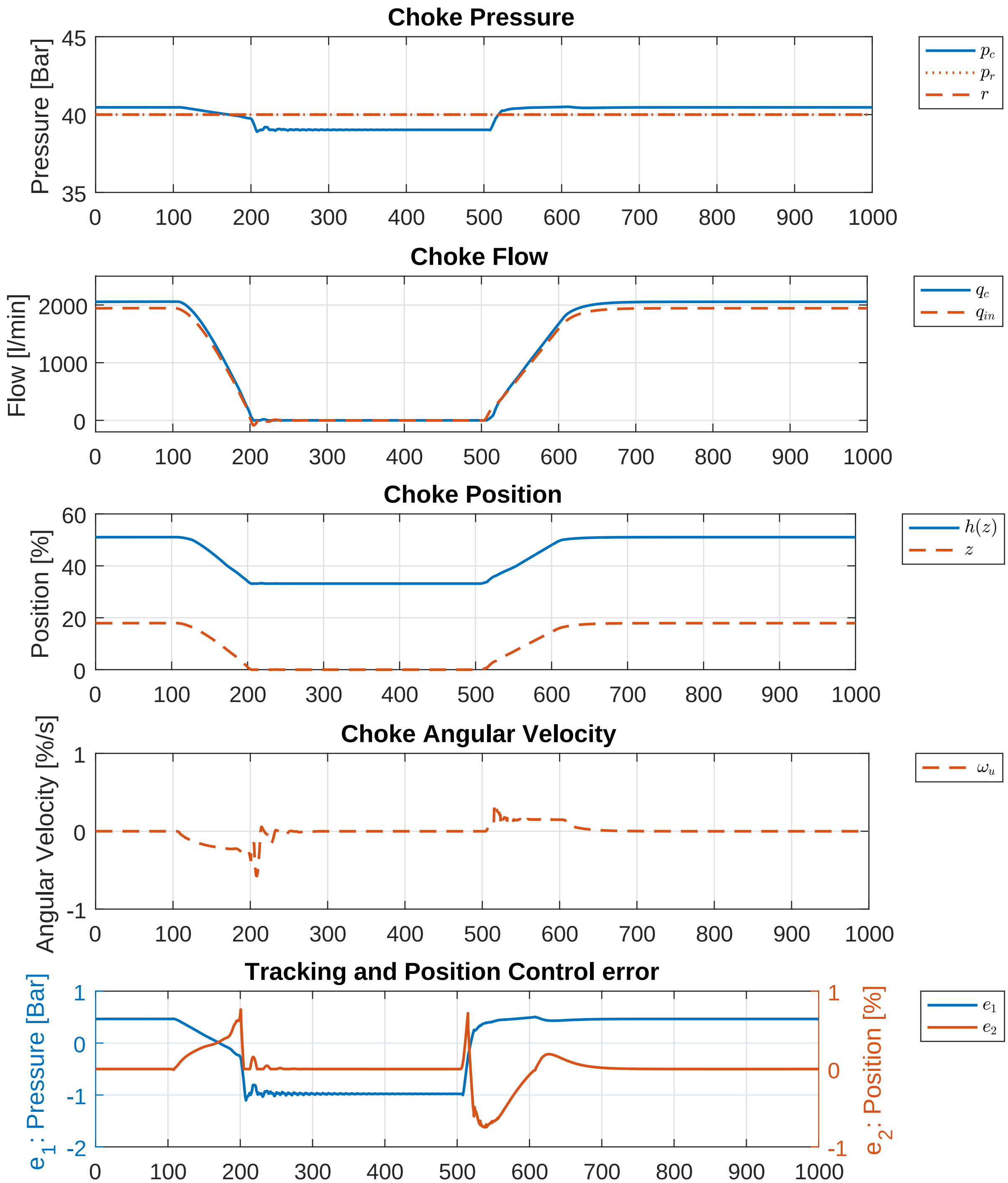

Figure B.4: Plot results from a simulation of the 2nd order integrator backstepping controller with Straume® in the connection scenario with high density in Straume

B.2 3rd Order Integrator Backstepping Controller

B.2.1 Reference value step changes scenario with low density

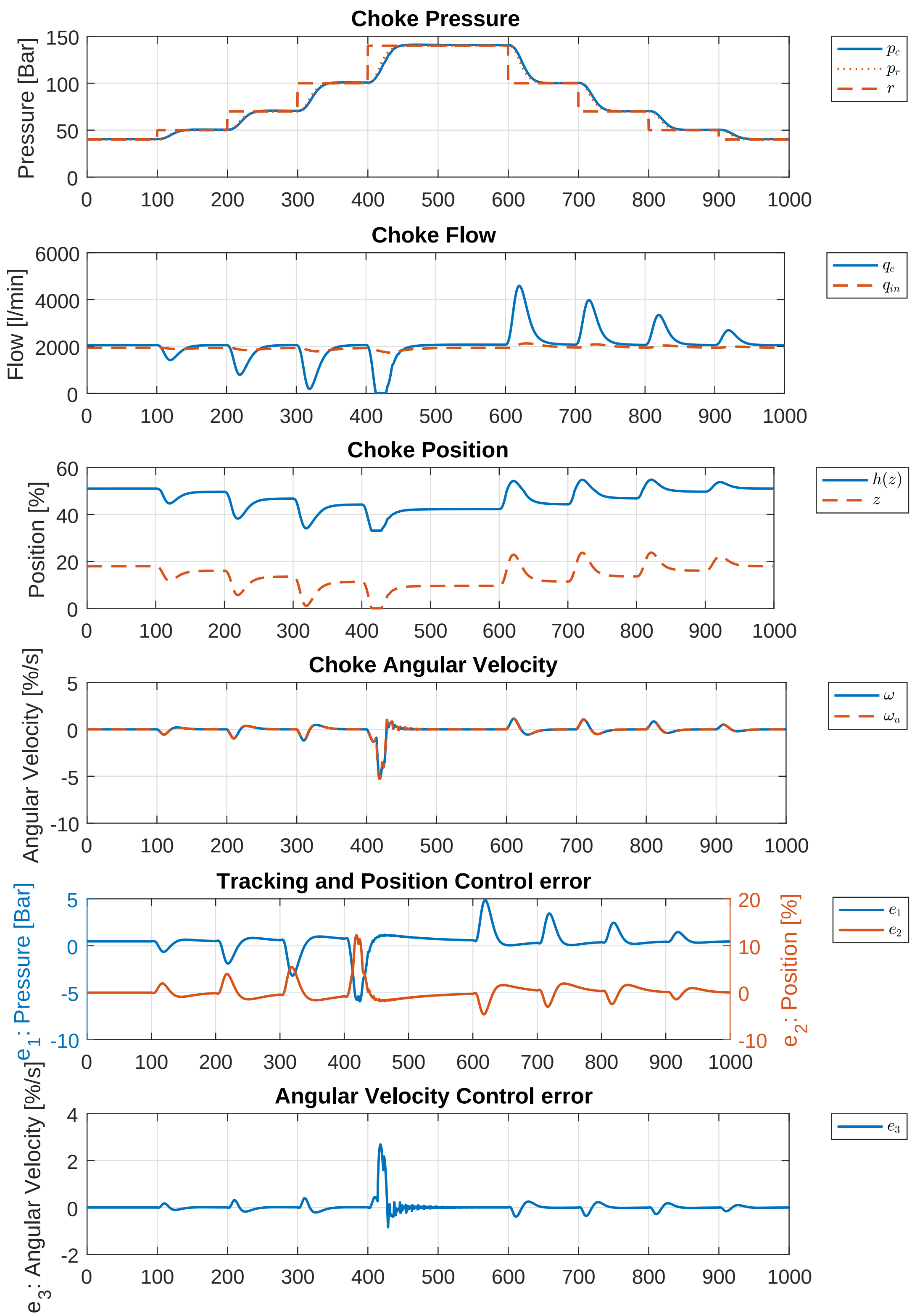

Figure B.5: Plot results from a simulation of the 3rd order integrator backstepping controller with Straume® in the p_r step changes scenario with low density in Straume

B.2.2 Reference value step changes scenario with high density

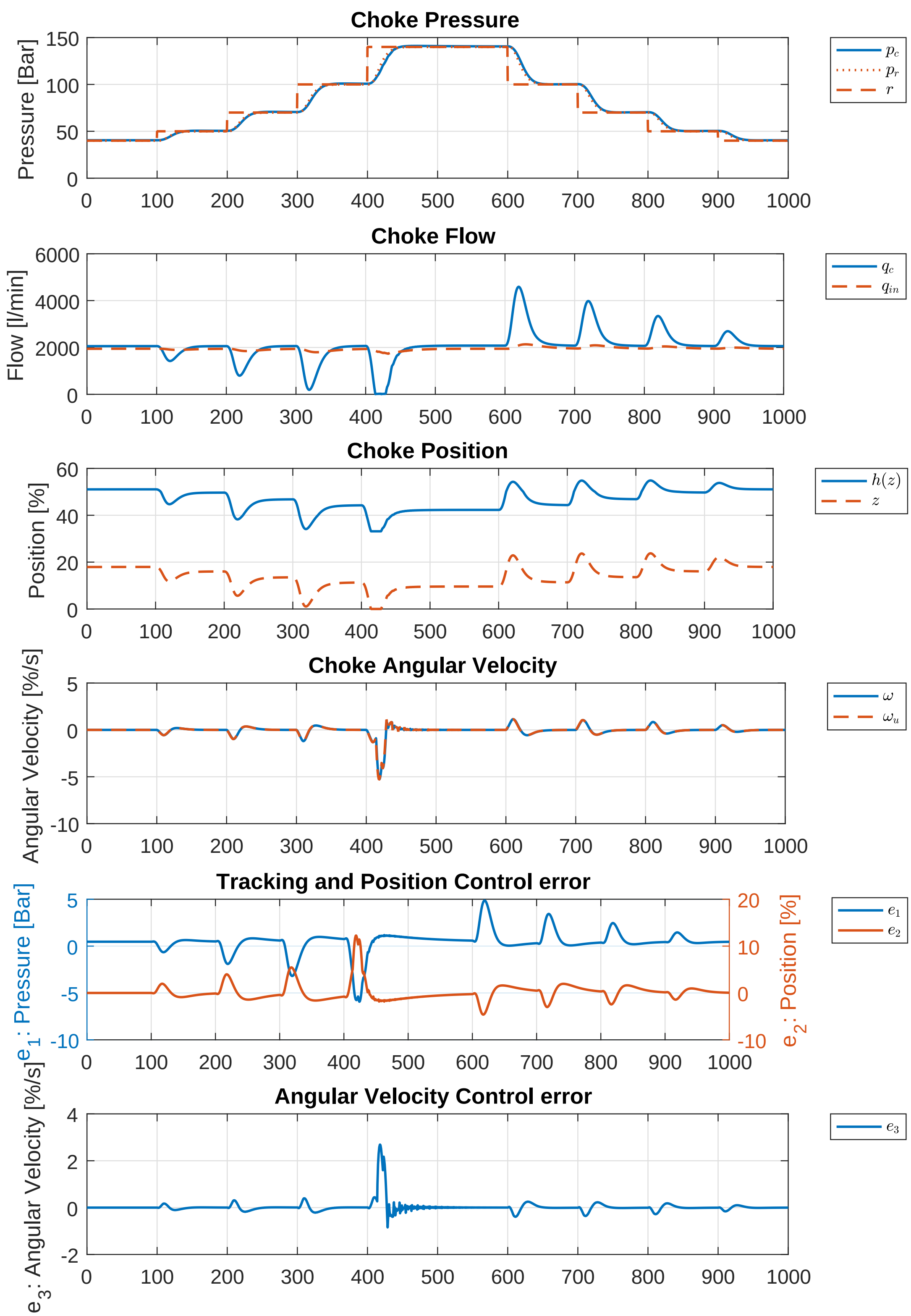

Figure B.6: Plot results from a simulation of the 3rd order integrator backstepping controller with Straume® in the p_r step changes scenario with high density in Straume

B.2.3 Connection scenario with low density

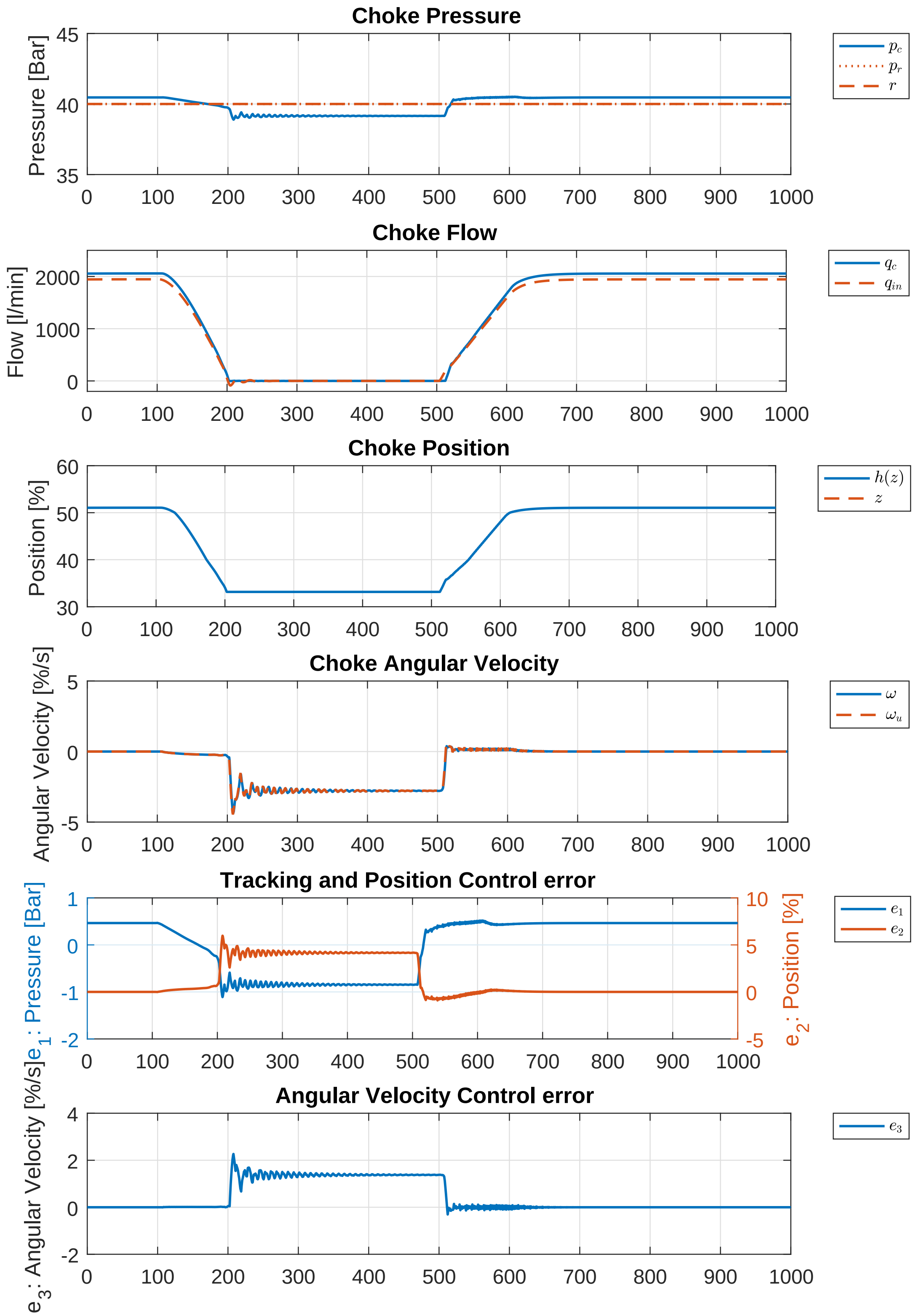

Figure B.7: Plot results from a simulation of the 3rd order integrator backstepping controller with Straume® in the connection scenario with low density in Straume

B.2.4 Connection scenario with high density

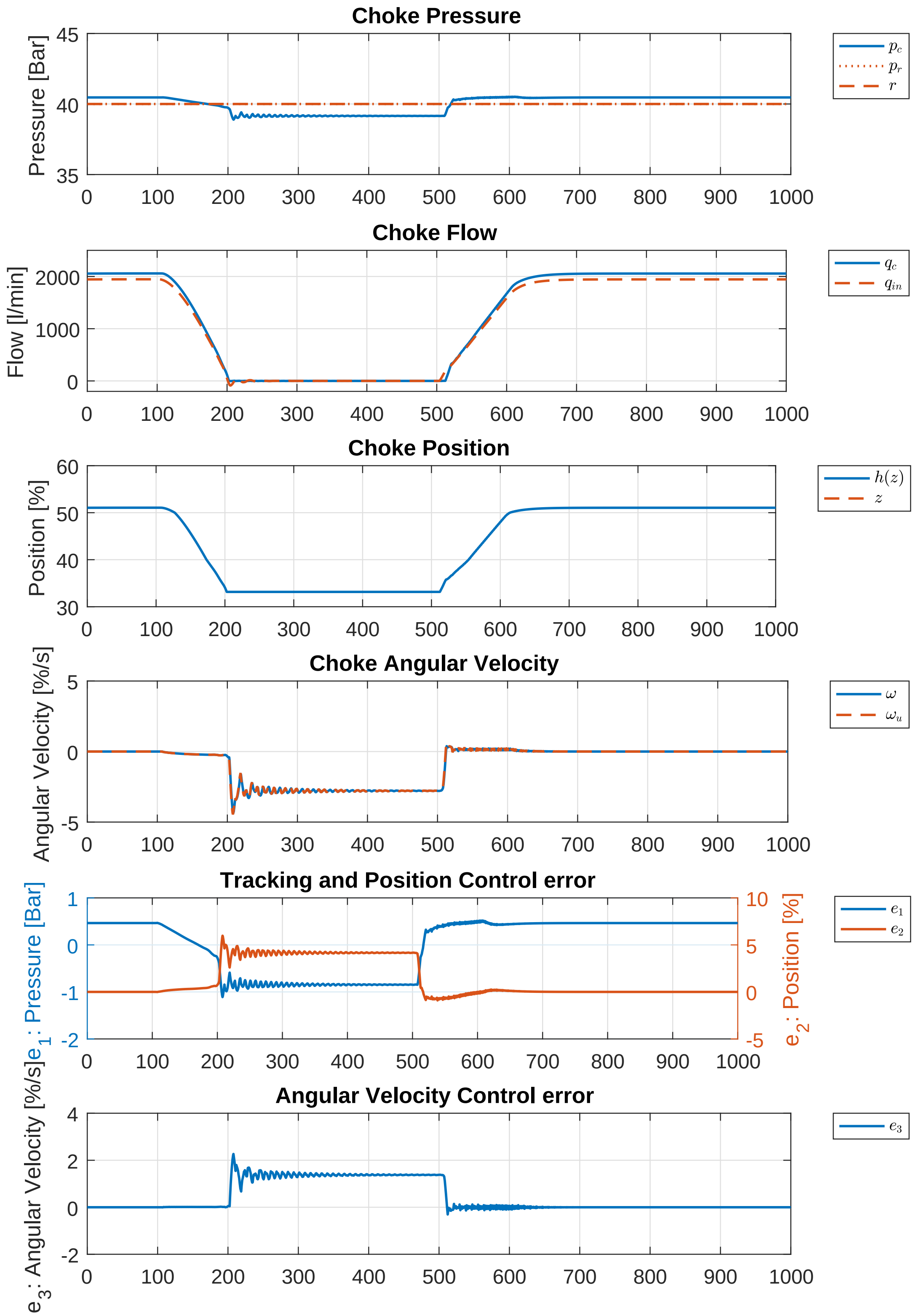

Figure B.8: Plot results from a simulation of the 3rd order integrator backstepping controller with Straume® in the connection scenario with high density in Straume

Bibliography

Khalil, H. K. (1996). *Noninear Systems.* Prentice-Hall, New Jersey.

Kokotovic, P. and Sussmann, H. (1989). A positive real condition for global stabilization of non-linear systems. *Systems & Control Letters*, 13(2):125–133.

Lamnabhi-Lagarrigue, F., Loría, A., and Panteley, E. (2005). *Advanced topics in control systems theory: lecture notes from FAP 2004*, volume 311. Springer Science & Business Media.

Saberi, A., Kokotovic, P., and Sussmann, H. (1990). Global stabilization of partially linear composite systems. *SIAM Journal on Control and Optimization*, 28(6):1491–1503.

Zhou, J. and Wen, C. (2008). *Adaptive backstepping control of uncertain systems: Nonsmooth nonlinearities, interactions or time-variations.* Springer.