Development of a Controller for a Servo-pneumatic System

Paul Christiaan Faria e Melo Dólleman

Supervisor: Prof. João Pedro Barata da Rocha Falcão Carneiro
Co-Supervisor: Prof. Fernando Gomes de Almeida

Faculdade de Engenharia da Universidade do Porto

October 2017

Acknowledgements

First and foremost, my sincere gratitude to my supervisor, Prof. João Falcão Carneiro, for guiding me throughout the last few months in the development of the work herein presented. His unrelenting support and patience with my questions (both big and small) were critical to the successful completion of this dissertation. I extend this gratitude to my co-supervisor, Prof. Fernando Gomes de Almeida, who was always available for the sometimes quite lengthy (and always insightful) discussions that happened throughout the semester.

I would also like to thank my parents and my sister for their continuous, unfaltering support, even at a distance, and for helping me keep perspective and relax when sometimes things did not go as expected.

Thank you to my close friends, both northern and southern, for their interest on what I was doing, and for providing a perfect counterbalance (one we all need) to the hours of work.

Abstract

Pneumatic actuators find widespread use in industry when motion between two end-points is required, given its high power to weight ratio and low maintenance requirements. However, classical PID control of pneumatic actuators may present several undesired features, like lack of motion smoothness and large steady-state errors. In this work, a two servo-valve architecture was developed for control of a servo-pneumatic system. In this architecture, the servo-valves are independently controlled - the one connected to the main chamber is controlled so as to maintain an approximately constant pressure of 3 bar in the auxiliary chamber, while the one connected to the auxiliary chamber handles motion control. By using this architecture with linear PID-family controllers, we aim to enhance motion smoothness and improve the steady-state errors usually obtained with PID controllers in classical architectures.

The first part of this work reviews and updates an existing mathematical model of a servo-pneumatic system. That model is linearised and represented in a manner that allows for the analysis of certain characteristics of the system, namely the coupling of pressure and velocity as control variables. This coupling is briefly studied, and it is seen how choked flow conditions can help in naturally decoupling pressure and velocity.

The control design process is presented, featuring heuristic tuning of control action terms that are introduced based on *a priori* knowledge of the system and experimental observations of certain problems, such as hunting limit-cycles and undesired pressure dynamics. In the end, the pressure loop features a PI controller with a reference velocity feed-forward term and a term that is proportional to the pressure in the main chamber. The motion loop features a proportional positioning controller in cascade with a P-D velocity controller with an additional term, proportional to the pressure error in the auxiliary chamber. A switching strategy was developed to make the transition between forward and backward controllers as smooth as possible.

This control architecture proved to be quite effective in tracking the general profile of an S-curve position input, therefore enabling smooth movements between positions. It also showed an acceptable step-response, although the tuning favoured the response to the aforementioned S-curve inputs. With that universal tuning, which also provided robustness to payload mass variations between 2 and 8 kg, a maximum steady-state positioning error of 0.45 mm was obtained, with no overshoot and overall smooth operation for movements of amplitude larger than 5 mm. A more specific tuning, which sacrificed robustness and step-response, enabled a decrease of the maximum steady-state error to 0.2 mm.

In direct comparison with a classical architecture PI-D controller, the developed controller generally presented significantly better performance, both in smoothness and accuracy.

Resumo

Os actuadores pneumáticos são amplamente usados na indústria para movimentações entre dois pontos, devido à sua relação potência/peso e à pouca manutenção necessária. No entanto, o controlo PID clássico desses actuadores apresenta alguns desafios, tais como pouca suavidade de movimento e erros de posicionamento bastante altos. Neste trabalho, foi desenvolvida uma arquitectura com duas servo-válvulas para controlo de um sistema servo-pneumático: a que está ligada à câmara principal é controlada de forma a manter a pressão na câmara auxiliar aproximadamente constante (3 bar), enquanto que a que está ligada à câmara auxiliar é responsável pelas especificações de movimento. Usando esta arquitectura com controladores da família PID, espera-se obter uma maior suavidade no movimento e uma melhoria dos erros de posicionamento normalmente associados a controladores PID de arquitectura clássica.

A primeira parte deste trabalho lida com uma revisão e actualização de um modelo matemático existente do sistema servo-pneumático. O modelo é depois linearizado e representado de uma forma que permite uma análise qualitativa de certas características do sistema, nomeadamente o acoplamento entre pressão e velocidade. Este acoplamento é alvo de análise, e é visto como a condição de caudal sónico permite um desacoplamento natural entre pressão e velocidade.

É apresentado o processo de desenvolvimento dos controladores, que envolve uma afinação heurística de termos de controlo introduzidos com base tanto em conhecimento *a priori* das dinâmicas do sistema, como em observações experimentais de problemas que surgem durante testes, como o aparecimento de ciclos-limite e dinâmicas de pressão indesejadas. A malha de pressão inclui um controlador PI com um termo de feed-forward da referência de velocidade, assim como um termo proporcional à pressão na câmara principal. Por sua vez, a malha de posicionamento é composta por um controlador proporcional de posição em cascata com um controlador de velocidade P-D, tendo este um termo adicional proporcional ao erro de pressão na câmara auxiliar. Foi ainda definida uma estratégia de comutação entre os controladores dos movimentos de avanço e recuo, de forma a que essa comutação aconteça o mais suavemente possível.

A arquitectura de controlo revelou-se eficaz no seguimento de uma solicitação de posição em S, conseguindo assim movimentos suaves entre posições. Mostrou ainda uma resposta ao degrau aceitável, apesar da afinação dos controladores favorecer a resposta a curvas em S. Com essa afinação universal, que também mostrou ser robusta a variações de carga entre 2 e 8 kg, foi obtido um erro máximo de posicionamento de 0.45 mm, sem sobre-elongação e com movimentos suaves entre posições, com excepção para movimentos de amplitude <5 mm. Uma afinação mais específica, obtida sacrificando robustez e qualidade de resposta ao degrau, mostrou uma melhoria dos erros de posicionamento para solicitações em S - o erro máximo foi de 0.2 mm, metade do obtido para a afinação universal. Em comparação directa com um controlador PI-D de arquitectura clássica, o controlador desenvolvido apresentou, regra geral, um desempenho significativamente melhor que o PI-D clássico, tanto em termos de suavidade como de precisão.

Table of contents

List of figures

List of tables

Nomenclature

Notation

$A_{A,B}$	Piston area in chamber A,B (m)
A_h	Cross-sectional area of the rod (m)
A_q	Heat-transfer area of the actuator chamber (m)
$C_{A1,A2,B1,B2}$	Sonic conductance of restriction 1,2 of servo-valve A,B ($\mathrm{m^3\,Pa^{-1}\,s^{-1}}$)
c_p	Specific heat for constant pressure ($\mathrm{J\,kg^{-1}\,K^{-1}}$)
c_v	Specific heat for constant volume ($\mathrm{J\,kg^{-1}\,K^{-1}}$)
ε	Error
ε_p	Pressure error (Pa)
ε_{ss}	Steady-state positioning error (m)
ε_x	Positioning error (m)
F	Force (N)
F_{ext}	External force (N)
F_{fr}	Friction force (N)
$G_{pA,pB}$	Pressure/mass-flow coefficient of chamber A,B ($\mathrm{kg\,Pa^{-1}\,s^{-1}}$)
$G_{uA,uB}$	Mass flow gain of servo-valve A,B ($\mathrm{kg\,V^{-1}\,s^{-1}}$)
k	Heat conductance coefficient ($\mathrm{W\,K^{-1}}$)
k_0	Heat conductance coefficient in equilibrium conditions ($\mathrm{W\,K^{-1}}$)
k_a	Viscous friction coefficient ($\mathrm{N\,s\,m^{-1}}$)
K_d	Proportional acceleration gain ($\mathrm{V\,m^{-1}\,s^{-2}}$)

K_{vel}	Proportional velocity gain ($\mathrm{V\,m^{-1}\,s^{-1}}$)
T_a	Anti-windup parameter (reset time) (s)
K_{ep}	Proportional ε_{paux} gain ($\mathrm{V\,Pa^{-1}}$)
K_{pa}	Proportional pressure gain ($\mathrm{V\,Pa^{-1}}$)
K_{pos}	Proportional positioning gain ($\mathrm{V\,m^{-1}}$)
K_v	Reference velocity feed-forward gain ($\mathrm{V\,m^{-1}\,s^{-1}}$)
λ	Heat transfer coefficient ($\mathrm{W\,m^{-2}\,K^{-1}}$)
λ_0	Equilibrium heat transfer coefficient ($\mathrm{W\,m^{-2}\,K^{-1}}$)
$\dot{m}$	Mass flow rate ($\mathrm{kg\,s^{-1}}$)
$\dot{m}_{A,B}$	Mass flow rate entering/exiting chambers A,B ($\mathrm{kg\,s^{-1}}$)
$\dot{m}_{A1,A2,B1,B2}$	Mass flow rate through restriction 1,2 of servo-valve A,B ($\mathrm{kg\,s^{-1}}$)
m	Payload Mass (kg)
N	Constant in the the derivative filter
n	Polytropic index
ρ	Density ($\mathrm{kg\,m^{-3}}$)
$p_{A,B}$	Pressure in chamber A,B (Pa)
$p_{A0,B0}$	Equilibrium pressure in chamber A,B (Pa)
p_{atm}	Atmospheric pressure (1 bar)
p_{aux}	Pressure in the auxiliary chamber (Pa)
p_s	Source pressure (7 bar)
$p_{ui,di}$	Upstream/downstream pressure in restriction i (Pa)
$\dot{Q}$	Heat transfer-rate (W)
R	Specific gas constant for air (ideal gas) ($\mathrm{J\,kg^{-1}\,K^{-1}}$)
r	Critical pressure ratio
$R_{1,2}$	Restriction 1,2 of a servo-valve
$T_{A,B}$	Temperature in chamber A,B (K)

$T_{A0,B0}$	Equilibrium temperature in chamber A,B (K)
$\tau_{A,B}$	Time constant of chamber A,B (s)
T_{amb}	Ambient air temperature (K)
T_d	Derivative time in the velocity controller (s)
T_{ip}	Integrator time in the pressure controller (s)
T_s	Source air temperature (K)
$u_{A,B}$	Control action applied to servo-valve A,B (V)
$u_{A0,B0}$	Equilibrium control action applied to servo-valve A,B (V)
u_{frz}	Integral control action at the instant in which the system enters the integrator freeze-zone
$V_{A,B}$	Volume of chamber A,B (m^3)
V_d	Dead Volume (m^3)
x	Piston position (m)
$\dot{x}$	Piston velocity ($m\,s^{-1}$)
$\dot{x}_{est}$	Estimated piston velocity ($m\,s^{-1}$)
$\dot{x}_{ref}$	Velocity reference ($m\,s^{-1}$)
x_0	Equilibrium position (x=0) (m)
x_{ref}	Position reference (m)
$\ddot{x}$	Piston acceleration ($m\,s^{-2}$)
$\ddot{x}_{est}$	Estimated piston acceleration ($m\,s^{-2}$)
z	Average deflection of the contact bristles for the LuGre model

Acronyms / Abbreviations

CLTF	closed-loop transfer function
P	Proportional
PID	Proportional, Integral, Derivative
PI-D	PID variant where differentiation is applied directly to the feedback signal, or the variable of interest is directly used (if available)

I-PD	PID variant where proportional gain and differentiation are applied directly to the feedback signals, or, for the -D term, the variable of interest is directly used (if available)
SMC	Sliding-Mode Control/Controller

Chapter 1

Introduction

1.1 Background and Motivation

Pneumatic actuation systems find widespread use throughout many industries, as they represent a mature, simple and readily available technology. These systems are economical, robust and compact, often being a competitive option for simple motion tasks involving mid-ranged speed and force. They rival DC servomotors in those applications, having the advantage of a higher power-to-weight ratio and the absence of heat or magnetic field generation. Furthermore, pneumatic systems are clean and low-maintenance, while also having a simple mechanical construction. These features make them a viable choice for specific applications such as wafer positioning devices (Fujita et al., 2002), robotic surgery in MRI environments (Wang et al., 2010), haptic devices (Le et al., 2011) or exoskeletons (Zhang et al., 2008).

There are, however, some disadvantages to the use of pneumatic systems. Besides the low energy-efficiency inherent to the use of compressed air, the highly non-linear behaviour of these systems (due to phenomena like friction) makes them difficult to model and control and, consequently, unsuitable for more complex control tasks involving smooth motion and accurate positioning. In order to tackle this problem, researchers all over the world have been using, in the recent years, advanced non-linear control strategies. However, given their complexity, these techniques may be difficult to implement and tune, and they can lead to very active control actions, which compromise motion smoothness.

This dissertation will try to address these issues by studying the implementation of linear controllers to control two servo-valves. One, connected to the main chamber, will be controlled in a way that guarantees approximately constant pressure in the opposite chamber. The other servo-valve will be controlled so as to guarantee the fulfilment of motion specifications.

The main reason for having pressure regulation in the auxiliary[1] chamber is motion smoothness, since a constant pressure of the right value will guarantee choked flow. In choked flow conditions, we remove the influence of the pressure differential on the volumetric flow-rate, which in turn defines piston velocity.

1.2 Main Goals

At the core of this dissertation is the development and evaluation of a two servo-valve architecture for the independent motion and pressure control of a servo-pneumatic system. The end-goal is to achieve, with that architecture, smooth and accurate motion control through the use of linear PID-family controllers. We wish to exploit the additional degree of freedom provided by the second servo-valve to not only have a motion controller but also a pressure regulator, in order to maintain an approximately constant pressure in the auxiliary chamber, with the aim of improving motion smoothness.

Including the work that precedes the control design and testing, the goals of this dissertation can be summarised in 4 main points:

- Review and update an existing model of the servo-pneumatic system, so as to make it a more accurate representation of reality;
- Linearise the model and represent it in a way that provides insight into some characteristics of the system, namely the coupling between the outputs of interest - pressure and velocity.
- Using the knowledge collected throughout the previous points, design pressure and motion controllers that take advantage of that knowledge to meet our qualitative requirements: motion smoothness and accurate positioning.
- Extensively test the controller performance along the full actuator stroke-length, its response to different kinds of inputs and the maximum steady-state errors it can guarantee. Compare the results with those of a PID-family controller with symmetrical control actions ($u_A = -u_B$).

[1] Auxiliary chamber: the chamber whose volume decreases during a movement between two end-points. Main chamber: the chamber whose volume increases during a movement between two end-points.

1.3 Positioning Control in Servo-Pneumatics - a brief state-of-the-Art

Despite the control-related drawbacks of pneumatic systems mentioned in Section 1.1, research has shown that it is possible, given the right control architectures, to achieve high-accuracy motion control both in positioning (Falcão Carneiro and Gomes de Almeida, 2012b) and trajectory following tasks (Falcão Carneiro and Gomes de Almeida, 2012a). The authors were able to achieve a control positioning error of ±5 µm, when arbitrarily positioning a piston of an actuator with 280 mm stroke length, and for payloads varying from 2.7 to 13 kg, therefore showing robustness. More recently, using an enhanced version of the controller developed in (Falcão Carneiro and Gomes de Almeida, 2012a), the authors were able to obtain even better results in the micrometer range (Falcão Carneiro and Gomes De Almeida, 2014). In fact, the control positioning error was limited only by the encoder resolution (1 µm), independently of the position, for loads of 3 and 8 kg, and with no controller re-tuning whatsoever. The actual positioning error was higher (±5 µm), although this was due to the accuracy of the encoder used.

The latest related published work by the same authors presented a controller architecture that included separate motion and pneumatic force controllers - the motion controller is based on integral sliding mode, while the force controller is based on a non-linear state feedback approach. The steady-state errors were shown to be under 50 µm (Carneiro and de Almeida, 2015). Even though these are higher than the previously mentioned values, the control law proved to be rather robust, withstanding a five-fold payload variation and, more importantly, maintaining good performance in the absence of a friction model. This is remarkable due to the importance of friction in industrial pneumatic behavior, given how hard it is to properly model it. Furthermore, the diversity of industrial pneumatic actuators makes it impossible to have a "universal" friction model - it is thus important to explore solutions that perform well in the absence of one.

Further results showing the positioning and tracking capabilities of properly controlled pneumatic actuators include those obtained by Bone and Ning (2007). Following the methodology proposed by Slotine and Li (1991), they designed a sliding mode controller which resulted in steady state errors of ±10 µm, with payloads varying from 2 to 11 kg. However, in this case the positioning error was found to vary with piston position.

Fok and Ong (1999) used an experimentally tuned PD controller to position a 40 mm bore size and 1.8 m stroke length rodless cylinder. Positioning errors of ±0.3 mm were obtained, in all five piston position references, for both 60 and 80 kg payloads. It is important to mention this result, because it was achieved through a PD approach instead of the sliding mode approach, the latter being common to most results herein

presented, and the first being more in line with the control strategies that will be studied during this dissertation. As we can see, many of these results were obtained using sliding mode approaches, which, while helping in dealing with model uncertainties, also introduce the drawback of excessive control activity, even with the use of smoothing strategies like boundary layers (Slotine and Li, 1991).

A relevant topic to the present study lies in the use of two servo-valves instead of one. Some of the previously mentioned studies use this configuration, but typically, servo-pneumatic systems use one five-orifice servo-valve to modulate the amount of air entering or exiting both cylinder chambers. Since there is only one spool, it is not possible to independently modulate the airflow through each working orifice. This means that the pressure dynamics of the two chambers are coupled in a manner that is dependent on the system state and also on the particular dimensions and characteristics of the servo-valves and actuator.

One important system property for pneumatic motion control is the pneumatic stiffness, and this is a property that can vary significantly with the system state, due to the already mentioned coupling of the pressure dynamics when using one servovalve. The pneumatic stiffness is dependent on the piston position and pneumatic force inside each chamber (Sorli and Gastaldi, 2009), and so, when using only one servovalve, it is also velocity-dependent. As shown by Carneiro and de Almeida (2013), this drawback can be very pronounced around zero velocity. This dependency is highly undesirable for control purposes, since sudden changes in the system's inherent bandwidth characteristics must be accounted for by the controller.

A two-servovalve architecture can surpass this drawback, enabling a decoupling of pressure dynamics and independent force control in each chamber (Carneiro and de Almeida, 2013). The particular approach presented in that work was developed for pneumatic force controllers, but it can be used as part of a cascaded motion controller - the author of this work expects that such a motion controller would present improved smoothness of movement, since pneumatic force variations would be kept under control.

Smaoui et al. (2005) used two servo-valves and a sliding mode approach, defining two sliding surfaces - one for the position of the piston and the other for the pressure of the main chamber. There are therefore two subsystems, one for motion control and the other for pressure control - this is similar to what will be studied in this dissertation, although not for a sliding mode approach. This architecture allowed the authors to choose the pressure of the main chamber for a given motion trajectory. In terms of results, positioning errors of approximately 0.2 mm were obtained, with the high control activity usually expected from sliding mode approaches.

In a more recent study, Girin et al. (2009) compared different control laws in the control of an electropneumatic system comprising two servovalves. The performance of two higher-order SMCs (HOSMCs) was assessed. One was a SISO (single input single

output), where the output was the piston position and the input was symmetrically applied to each servo-valve. The other was a MIMO (multiple input multiple output), with the output variables being the piston position and the pressure in one of the chambers - this architecture, along with the use of two servo-valves, enables an improvement regarding the already discussed pneumatic stiffness of the system. The SISO controlled system showed positioning errors of several tens of micrometers, depending on the position reference and payload. The MIMO controlled system showed even better results, with positioning errors of few tens of micrometers. Even though HOSMCs typically reduce chattering, the MIMO still presented significant levels of control activity.

The results herein presented often show errors in the range of tens or hundreds of micrometers, providing a general idea of the possibilities in accurate motion control of pneumatic systems. It seems that sliding-mode approaches result in the lowest positioning errors, with the added advantage of robustness in the face of uncertainties and payload variations. This comes at the cost of excess control activity and difficulty in achieving smooth motion. This dissertation will avoid the sliding-mode approach, focusing on linear controllers and the use of two servo-valves to independently control pressure in one chamber and motion in the other.

1.4 Outline of the Thesis

This dissertation is divided into 5 chapters, the present one being **Chapter 1**.

Chapter 2 will deal with the review and update of the mathematical model of the servo-system. A 6th-order non-linear model is presented and subsequently reduced to a 4-th order non-linear model. Resorting to a Taylor-expansion, that reduced-order model is linearised around an equilibrium point. The linearisation allows for state-space and transfer-function representations of the system, which provide some insight into certain characteristics of the system, such as the coupling between pressure and velocity - the controlled variables. The nature of that coupling is briefly studied, along with an analysis of how choked-flow conditions allow for a decoupling of those variables. The end of the chapter deals with the addition of certain elements to the MATLAB® implementation of the system model, so as to make it a more accurate representation of a real servo-pneumatic set-up.

Chapter 3 will first present a classical PID controller, to serve as a benchmark with which we can compare the results of the developed controllers (Chapter 4). The following sections will then explain the control design process. With the insight from Chapter 2, and responding to specific problems (some expected, others not) that arose during the development phase, the controllers are gradually built through educated additions of control action terms and heuristic tuning, with the goal of having pressure and motion controllers which satisfy the requirements of smooth motion and accurate positioning. Because piston velocity and acceleration values are not directly available for use by the controllers, a section is dedicated to an overview of the Kalman filter and how it can be used as a velocity and acceleration estimator. The final section presents a classical architecture PI-D controller, to be used as a benchmark with which we can compare the results of the developed controllers (Chapter 4).

Chapter 4 presents testing and performance analysis of the controllers obtained in Chapter 3. Step-response and S-curve response are studied, and a brief robustness analysis is done, with payload mass as the variable parameter. The developed controllers are then more comprehensively tested in a full sweep of the actuator-stroke length, with movements of varying amplitudes.

Chapter 5 presents the conclusions that can be drawn from this work, and discusses further paths of improvement for this type of control architecture in servo-pneumatics.

Chapter 2

System Model

2.1 Experimental set-up to be simulated

This section will introduce the experimental set-up that is to be modelled. Since this work is fully done in a simulation environment, only the relevant aspects of the set-up are mentioned. For example, air treatment is a part of the set-up that is not relevant for modelling purposes, but it should be noted that in the simulation environment, p_s can be assumed constant without loss of accuracy because in reality, there is a 5 litre accumulator that compensates for p_s fluctuations.

A photograph of the experimental installation can be seen in Fig. 2.1.

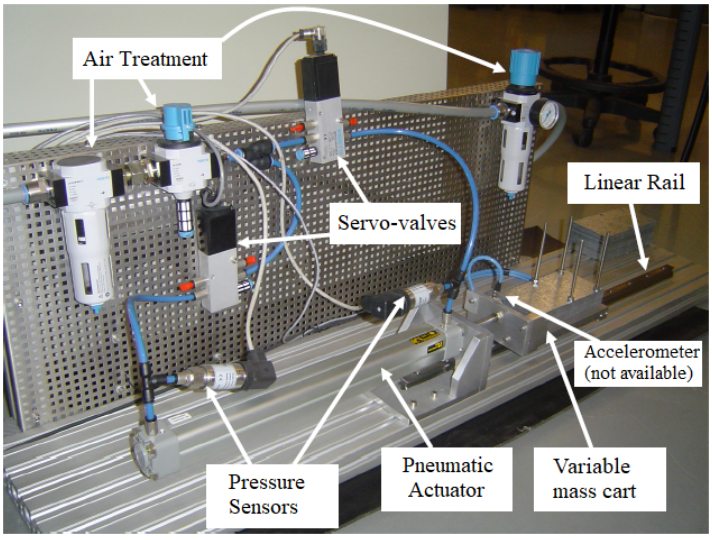

Fig. 2.1 Photograph of the servo-pneumatic system to be modelled (Falcão Carneiro, 2007)

Servo-valves

In order to modulate the air flow entering/exiting the actuator, a servo-valve is connected to each chamber. These are five-way, closed-center servo-valves manufactured by FESTO with the reference MPYE-5-1/8-HF-010-B. Since a three-way servo-valve is sufficient for flow control in an actuator chamber, the remaining two orifices are sealed. Table 2.1 presents the general servo-valve characteristics.

Table 2.1 Servo-valve Characteristics (ASCO-Joucomatic, 2011)

Characteristic	Value
Max. admissible pressure	10 bar
Supply Voltage	24 V
Nominal Flow	700 SLPM
Bandwidth (-3 dB)	100 Hz
Reference Voltage	[0 10] V

Actuator and Cart

The actuator modelled in this work is a double-acting, asymmetric pneumatic actuator manufactured by *ASCO®Joucomatic*, with *Omega* type seals and tunable internal damping (which was deactivated) (ASCO-Joucomatic, 2011). The characteristics of the actuator are presented in Table 2.2.

Table 2.2 Actuator Characteristics (ASCO-Joucomatic, 2011)

Characteristic	Value	Description
D_p	0.032 m	Piston diameter
D_h	0.012 m	Rod diameter
l	0.4 m	Stroke length
A_A	$8.0425\text{x}10^{-4}$ m^2	Piston area in chamber A
A_B	$6.9115\text{x}10^{-4}$ m^2	Piston area in chamber B
A_h	$1.1310\text{x}10^{-4}$ m^2	Cross-sectional rod area

The dead volumes of each actuator chamber are assumed equal and given by (Falcão Carneiro, 2007):

$$V_{dA} = V_{dB} = 5.15\text{x}10^{-5}\ \text{m}^3$$

The cart, attached to the cylinder rod, is guided by a linear guide-rail and can be loaded with variable payload masses.

Sensors and Data Acquisition

The position measurements are obtained by use of a digital encoder with a resolution of 5 µm, while the pressure readings are obtained through pressure sensors for each chamber, which are subject to electrical noise (to be discussed later). These sensors, being analogue, have null resolution, but since they are obtained by an A/D converter, the resolution of the converter is imposed on the pressure signals. However, the resulting quantization of the pressure values is negligible. The pressure readings are then passed through a low-pass filter with 100 Hz cut-off frequency. The sampling frequency of the data acquisition system is 1000 Hz.

2.2 Model Overview

This chapter will present the mathematical models used to describe the behaviour of the different system properties. A servo-valve model provides the mass flow-rates that go into the actuator chambers, which in turn have their temperature and pressure given by a thermodynamic model. A mechanical model predicts the motion of the system, based on the pneumatic forces acting on the piston. Friction must of course be taken into account, and so it is given by a LuGre friction model (de Wit et al., 1995). Fig. 2.2 illustrates the basic relationship between the different models.

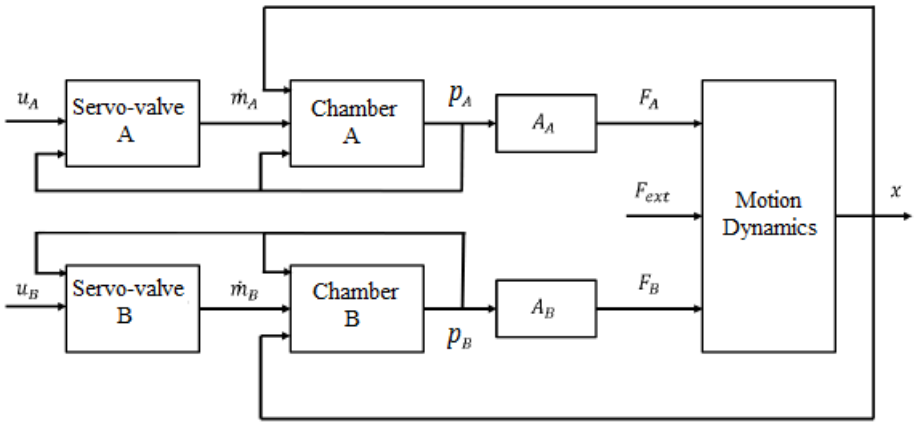

Fig. 2.2 Diagram of the main system blocks, adapted from (Ferreira da Silva, 2015)

After having the complete system model, some simplifications are made and a reduced-order model is presented that is more suitable for control purposes, as shown by Falcão Carneiro (2007). That model is then linearised through a Taylor-series expansion, resulting in a 4th-order linear system model.

The structure of the linearised model sheds some light into certain characteristics of the system, namely the coupling between the initial variables of interest - pressure and velocity (and later position). This coupling is studied, and an analysis is made of the conditions in which it can be considered negligible.

The last section deals with the *Simulink*® implementation of the system model, along with the addition of "real-life" elements like electrical noise in the pressure measurements and the quantization of the position readings.

2.3 Pneumatic Actuator Model

We now proceed to the modelling of the actuator dynamics. A mechanical model is presented, based on Newton's 2nd law, that provides the acceleration of the piston due to the forces acting on it. A thermodynamic model describes the evolution of pressure and temperature inside the actuator chambers, and a LuGre model provides the dissipative friction force. The combination of these models, along with the servo-valve mass flow rate equations, make up the complete, non-linear system model.

2.3.1 Mechanical Model

The mechanical model of the actuator can be easily found through direct application of Newton's 2nd law. Let us consider the schematics of Fig. 2.3.

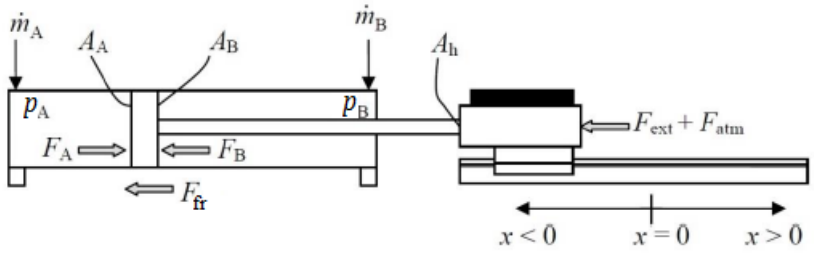

Fig. 2.3 Schematic representation of the pneumatic actuator (Falcão Carneiro (2007))

F_A and F_B are the pneumatic forces acting on each face of the piston, with areas A_A and A_B. A_h is the cross-sectional area of the piston rod. F_{fr} is the friction force, F_{ext} represents any external force acting on the payload, and F_{atm} is the force due to atmospheric pressure acting on the piston rod.

Applying Newton's 2nd law to the system represented in Fig. 2.3 and noting that m represents the payload mass, we get:

$$m\ddot{x} = F_A - F_B - F_{atm} - F_{fr} - F_{ext} \tag{2.1}$$

where

$$F_A = p_A A_A \tag{2.2}$$

$$F_B = p_B A_B \tag{2.3}$$

$$F_{atm} = p_{atm} A_h \tag{2.4}$$

And we can finally describe the motion dynamics with the following equations:

$$m\ddot{x} = p_A A_A - p_B A_B - p_{atm} A_h - F_{fr} - F_{ext} \tag{2.5}$$

$$\frac{dx}{dt} = \dot{x} \tag{2.6}$$

The model used to obtain the friction force is presented in the following subsection.

Friction Model

Friction plays an important role in pneumatics - it is a relatively complex phenomenon that poses a challenge to the precise control of position and velocity, especially in the realm of linear control. It is therefore important to have an accurate friction model if one wants to properly describe the dynamics of a servo-pneumatic system. Unfortunately, friction is quite hard to model, and most classical friction models, such as Coulomb and viscous friction, do not satisfactorily describe what happens for low velocities or when crossing zero velocity (de Wit et al., 1995).

The aforementioned types of friction can be combined with the *Stribeck* effect to better describe low-velocity friction forces. The *Stribeck* effect is a phenomenon observed at low velocity in which the friction force decreases after the initial relative movement of the contact surfaces, only to rise up again as the velocity increases. The *Stribeck* effect allows for the prediction of stick-slip motion, a prevalent phenomenon in servo-pneumatics. As such, in the context of this work, a pre-requisite of the friction model is that it be able to predict stick-slip motion.

All the aforementioned models, even though they capture important properties of friction, represent static relationships between velocity and friction, and as such fail to account for time-dependent (dynamic) behaviours - these are "memory-less" models

(de Wit et al., 1995). The Dahl model (Dahl, 1968) is a dynamic model that addresses this, but while being able to capture some dynamic properties of friction, it does not include the *Stribeck* effect, thus failing to predict stick-slip motion.

This brings us to the LuGre model. Introduced by de Wit et al. (1995), its name is derived from the universities in which it was developed (Lund in Sweden and Grenoble in France). The LuGre model is deemed the strongest candidate for this work, since it is a dynamic model that captures the *Stribeck* effect, Coulomb friction, static friction (stiction) and viscous friction. The characteristic friction force-velocity curve of the LuGre model is presented in Fig. 2.4.

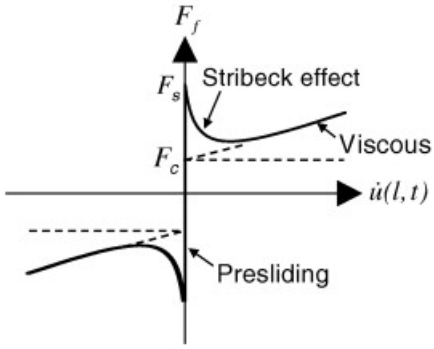

Fig. 2.4 Characteristic friction force-velocity curve of the LuGre model (Fung et al. (2008))

In the LuGre model, the friction interface between two surfaces is thought of as a contact between elastic bristles. Fig. 2.5 illustrates that interface, but with the lower bristles shown as rigid for the sake of simplicity.

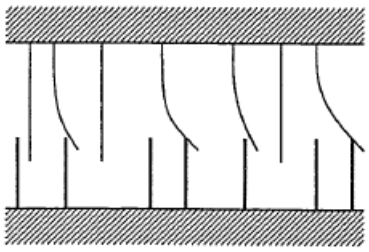

Fig. 2.5 Friction interface according to the LuGre model (de Wit et al. (1995))

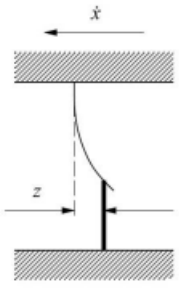

Fig. 2.6 Average bristle deflection (Pereira, 2013)

When a tangential force is applied, the bristles deflect like springs, giving rise to the friction force. If the tangential force is sufficiently large, the deflection will eventually lead to slip and there will be movement. The model assumes that the behaviour of one bristle represents the average behaviour of all bristles (Fig 2.6).

We now refer to Åström and de Wit (2008) and de Wit et al. (1995) to formally present the model. The average deflection of the bristles is denoted by z, the dynamics of which are modelled by

$$\frac{dz}{dt} = \dot{x} - \sigma_0 \frac{|\dot{x}|}{g(\dot{x})} z \tag{2.7}$$

where $\dot{x}$ is the relative velocity between the contact surfaces, σ_0 describes the stiffness of the bristles, and $g(\dot{x})$ is a function that accounts for the *Stribeck* effect and Coulomb friction. It is given by

$$g(\dot{x}) = \alpha_0 + \alpha_1 e^{(\dot{x}/\nu_s)^2} \tag{2.8}$$

where α_0 represents Coulomb friction and α_1 the *Stribeck* effect, with ν_s being the so-called *Stribeck* velocity.

The total friction force predicted by the model is then given by

$$F_{fr} = \sigma_0 z + \sigma_1 \dot{z} + f(\dot{x}) \tag{2.9}$$

where σ_1 is a damping coefficient associated with bristle micro-displacement, and $f(\dot{x})$ accounts for viscous friction. Typically (and in this work), $f(\dot{x})$ is taken simply as $f(\dot{x}) = \sigma_2 \dot{x}$ and σ_1 is a velocity-dependent term (Olsson, 1996), given by

$$\sigma_1 = \sigma_1' e^{-(\dot{x}/\nu_c)^2} \tag{2.10}$$

where σ_1' is the pre-sliding damping coefficient, and ν_c the velocity value beyond which damping starts to decrease.

All the aforementioned parameters were determined, for the pneumatic actuator discussed in this work, in a more extensive study of the LuGre model done by Pereira (2013). They are presented in Table 2.3.

Table 2.3 Parameters of the LuGre model, as determined by Pereira (2013)

Parameter	Value	Description
α_0	19.9850 N	Coulomb Friction
α_1	18.4950 N	*Stribeck* Effect
ν_s	0.0077 $\mathrm{m\,s^{-1}}$	*Stribeck* Velocity
σ_0	32862500 $\mathrm{N\,m^{-1}}$	Bristle Stiffness
σ_1'	4.1547x10^{-4} $\mathrm{N\,s\,m^{-1}}$	Pre-sliding damping coefficient
ν_c	0.0024 $\mathrm{m\,s^{-1}}$	Velocity beyond which damping decreases
σ_2	76.6300 $\mathrm{N\,s\,m^{-1}}$	Viscous Friction

2.3.2 Thermodynamic Model of the Actuator Chambers

The thermodynamic model aims to describe the pressure and temperature dynamics in the actuator chambers. The model herein presented was obtained by application of the Reynolds transport theorem (White, 2011) to the control volume of an actuator chamber (see Fig. 2.7). For the full description of the model and its deduction, see Falcão Carneiro (2007).

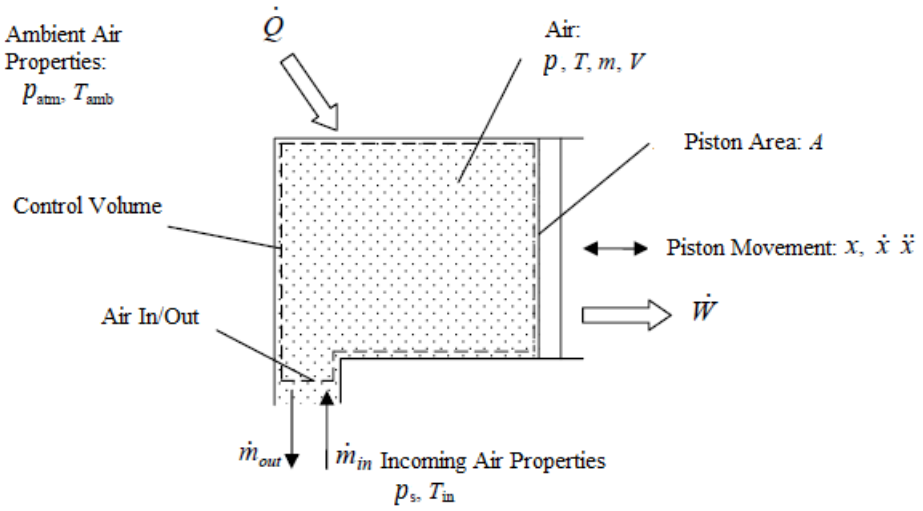

Fig. 2.7 Schematic representation of a pneumatic actuator chamber (adapted from Falcão Carneiro (2007))

From the aforementioned work by Falcão Carneiro (2007), we know that the pressure and temperature dynamics in a pneumatic actuator chamber are given by:

$$\frac{dp}{dt} = -\gamma\frac{p}{V}\frac{dV}{dt} + \gamma\frac{R}{V}\dot{m}_{in}T_{in} - \gamma\frac{R}{V}\dot{m}_{out}T + \frac{\gamma-1}{V}\dot{Q} \tag{2.11}$$

$$\frac{dT}{dt} = \frac{T}{V}\frac{dV}{dt}(1-\gamma) + \dot{m}_{in}\frac{RT}{Vp}(\gamma T_{in} - T) - \dot{m}_{out}\frac{RT^2}{Vp}(\gamma-1) + \frac{(\gamma-1)T}{pV}\dot{Q} \tag{2.12}$$

where $\gamma = c_p/c_v$, c_p being the specific heat constant of air at constant pressure, and c_v the specific heat constant at constant volume. R is the specific gas constant for air. If heat exchanges through radiation are neglected and the actuator wall temperature is assumed equal to the ambient temperature T_{amb}, the heat-transfer term $\dot{Q}$ can be given by equation 2.13.

$$\dot{Q} = \lambda A_q(x)(T_{amb} - T) \tag{2.13}$$

where $A_q(x)$ is the heat-transfer area of the actuator chamber and λ is the heat-transfer coefficient. It is common to use the simplified Eichelberg model (Eichelberg, 1939) to establish λ as a function of just pressure and temperature:

$$\lambda(p,T) = \lambda_0\sqrt{\left(\frac{pT}{p_0T_0}\right)} \tag{2.14}$$

in which λ_0 is a heat-transfer coefficient for p_0, T_0 - the reference values for pressure and temperature, respectively (Falcão Carneiro, 2007). Equations 2.11 and 2.12 can now be rewritten as:

$$\frac{dp}{dt} = -\gamma\frac{p}{V}\frac{dV}{dt} + \gamma\frac{R}{V}\dot{m}_{in}T_{in} - \gamma\frac{R}{V}\dot{m}_{out}T + \frac{\gamma-1}{V}\lambda_0\sqrt{\left(\frac{pT}{p_0T_0}\right)}A_q(x)(T_{amb} - T) \tag{2.15}$$

$$\begin{aligned}\frac{dT}{dt} = \frac{T}{V}\frac{dV}{dt}(1-\gamma) + \dot{m}_{in}\frac{RT}{Vp}(\gamma T_{in} - T) - \dot{m}_{out}\frac{RT^2}{Vp}(\gamma-1) \\ + \frac{(\gamma-1)T}{pV}\lambda_0\sqrt{\left(\frac{pT}{p_0T_0}\right)}A_q(x)(T_{amb} - T)\end{aligned} \tag{2.16}$$

2.4 Servo-valve Model

2.4.1 Servo-valve Modelling according to ISO 6358

The servo-valve model is obtained by use of the ISO 6358 standard. Let us consider the 3-way servo-valve represented in Fig. 2.8

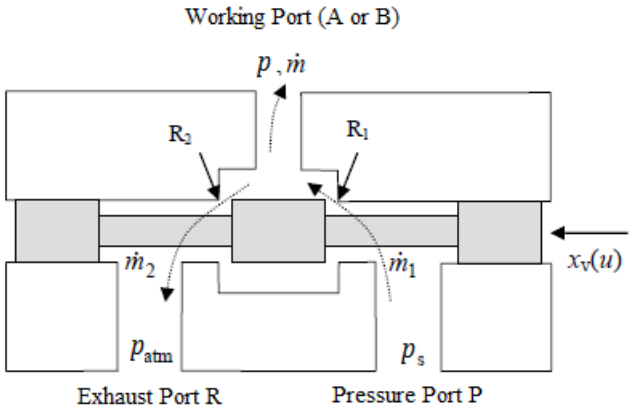

Fig. 2.8 Schematic representation of a 3-way servo-valve (Falcão Carneiro (2007))

Port P is connected to the pressure source and Port R to the atmosphere. The working port is connected to an actuator chamber (A or B). The area of restrictions R_1 and R_2 varies with the spool position x_v, which is a function of a voltage u (the control action). The amount of air entering or leaving the actuator chambers is thus modulated by the spool position, which in turn is defined by the control action. For the sake of clarity, it should be noted that this means that a variable dependent of u is, perhaps more intuitively, dependent on the *position* of the valve spool, and thus on the area of the restriction. The spool position x_v is assumed to have a static relationship with the control action u, since the servo-valve dynamics are much faster than the actuator dynamics. This is a common simplification in servo-pneumatics - the servo-valves typically present a bandwidth of 100 Hz (FESTO, 2005), while for the actuator it lies around 6 Hz (Falcão Carneiro, 2007).

According to ISO 6358, for a given restriction R_i, the mass flow rate through that restriction is given by:

$$\dot{m} = C_i(u) p_{ui} \rho_0 \sqrt{\frac{293.15}{T_{ui}}} Y \tag{2.17}$$

with Y given by:

$$Y = \begin{cases} 1, & \text{if } \dfrac{p_{di}}{p_{ui}} \leq r_i(u) \\ \sqrt{1 - \left(\dfrac{p_{di}/p_{ui} - r_i}{1 - r_i}\right)^2}, & \text{if } \dfrac{p_{di}}{p_{ui}} > r_i(u) \end{cases} \tag{2.18}$$

where C_i is the sonic conductance, p_{ui} and p_{di} are, respectively, the upstream and downstream pressures, and r_i is the critical pressure ratio of the restriction, below which the flow becomes choked. $C_i(u)$ and $r_i(u)$ are given as functions of u because they vary with the area of the restriction. Their dependence of u was determined experimentally, for the servo-valves discussed in this work, by Falcão Carneiro (2007). Throughout this text, their dependence of u will be omitted in the notation - for the sake of simplicity, they will be represented only as C_i and r_i.

Let us now refer once again to Fig. 2.8. We can easily observe that, by the law of conservation of mass, the mass flow rate entering through pressure port P must be equal to the sum of the mass flow rates through working port A (or B) and the exhaust port R. We can then take the net mass flow rate through the working port as being the algebraic sum of the flow rates through the restrictions R_1 and R_2. We can then write:

$$\dot{m}_A = \dot{m}_{A1} - \dot{m}_{A2} \tag{2.19}$$

$$\dot{m}_B = \dot{m}_{B1} - \dot{m}_{B2} \tag{2.20}$$

At this point we break away from the literature to introduce some changes that were made to the servo-valve model presented by Falcão Carneiro (2007).

Updated servo-valve model: reverse-flow

The servo-valve model herein introduced will be an updated version of the one presented by Falcão Carneiro (2007) - it will account for reverse flow in the servo-valve restrictions, in case the pressure in an actuator chamber is either greater than p_s or smaller than p_{atm}.

We must therefore distinguish between working conditions to define the mass flow rates in each restriction. We then have, for servo-valve A:

If $p_s \geq p_A$ (normal flow in restriction 1):

$$\dot{m}_{A1} = C_{A1} p_s \rho_0 \sqrt{\frac{293.15}{T_s}} \begin{cases} 1, & \text{if } \dfrac{p_A}{p_s} \leq r_{A1} \\ \sqrt{1 - \left(\dfrac{p_A/p_s - r_{A1}}{1 - r_{A1}} \right)^2}, & \text{if } \dfrac{p_A}{p_s} > r_{A1} \end{cases} \tag{2.21}$$

If $p_s < p_A$ (reverse flow in restriction 1):

$$\dot{m}_{A1} = -C_{A1} p_A \rho_0 \sqrt{\frac{T_s}{293.15}} \begin{cases} 1, & \text{if } \dfrac{p_s}{p_A} \leq r_{A1} \\ \sqrt{1 - \left(\dfrac{p_s/p_A - r_{A1}}{1 - r_{A1}} \right)^2}, & \text{if } \dfrac{p_s}{p_A} > r_{A1} \end{cases} \tag{2.22}$$

If $p_A \geq p_{atm}$ (normal flow in restriction 2):

$$\dot{m}_{A2} = C_{A2} p_A \rho_0 \sqrt{\frac{293.15}{T_A}} \begin{cases} 1, & \text{if } \dfrac{p_{atm}}{p_A} \leq r_{A2} \\ \sqrt{1 - \left(\dfrac{p_{atm}/p_A - r_{A2}}{1 - r_{A2}} \right)^2}, & \text{if } \dfrac{p_{atm}}{p_A} > r_{A2} \end{cases} \tag{2.23}$$

If $p_A < p_{atm}$ (reverse flow in restriction 2):

$$\dot{m}_{A2} = -C_{A2} p_{atm} \rho_0 \sqrt{\frac{T_A}{293.15}} \begin{cases} 1, & \text{if } \dfrac{p_A}{p_{atm}} \leq r_{A2} \\ \sqrt{1 - \left(\dfrac{p_A/p_{atm} - r_{A2}}{1 - r_{A2}} \right)^2}, & \text{if } \dfrac{p_A}{p_{atm}} > r_{A2} \end{cases} \tag{2.24}$$

We then proceed similarly for servo-valve B.

If $p_s \geq p_B$ (normal flow in restriction 1):

$$\dot{m}_{B1} = C_{B1} p_s \rho_0 \sqrt{\frac{293.15}{T_s}} \begin{cases} 1, & \text{if } \dfrac{p_B}{p_s} \leq r_{B1} \\ \sqrt{1 - \left(\dfrac{p_B/p_s - r_{B1}}{1 - r_{B1}} \right)^2}, & \text{if } \dfrac{p_B}{p_s} > r_{B1} \end{cases} \tag{2.25}$$

If $p_s < p_B$ (reverse flow in restriction 1):

$$\dot{m}_{B1} = -C_{B1} p_B \rho_0 \sqrt{\frac{T_s}{293.15}} \begin{cases} 1, & \text{if } \dfrac{p_s}{p_B} \leq r_{B1} \\ \sqrt{1 - \left(\dfrac{p_s/p_B - r_{B1}}{1 - r_{B1}} \right)^2}, & \text{if } \dfrac{p_s}{p_B} > r_{B1} \end{cases} \tag{2.26}$$

If $p_B \geq p_{atm}$ (normal flow in restriction 2):

$$\dot{m}_{B2} = C_{B2} p_B \rho_0 \sqrt{\frac{293.15}{T_B}} \begin{cases} 1, & \text{if } \dfrac{p_{atm}}{p_B} \leq r_{B2} \\ \sqrt{1 - \left(\dfrac{p_{atm}/p_B - r_{B2}}{1 - r_{B2}} \right)^2}, & \text{if } \dfrac{p_{atm}}{p_B} > r_{B2} \end{cases} \tag{2.27}$$

If $p_B < p_{atm}$ (reverse flow in restriction 2):

$$\dot{m}_{B2} = -C_{B2} p_{atm} \rho_0 \sqrt{\frac{T_B}{293.15}} \begin{cases} 1, & \text{if } \dfrac{p_B}{p_{atm}} \leq r_{B2} \\ \sqrt{1 - \left(\dfrac{p_B/p_{atm} - r_{B2}}{1 - r_{B2}} \right)^2}, & \text{if } \dfrac{p_B}{p_{atm}} > r_{B2} \end{cases} \tag{2.28}$$

Equations 2.4 through 2.11 completely describe the mass flow rates through both restrictions of both servo-valves, for normal and reverse flow conditions.

2.5 Non-Linear System Model

The mechanical and servo-valve models, in conjunction with the thermodynamic model of the actuator chambers, make up the complete 6th-order non-linear model of the servo-pneumatic system:

$$\frac{dp_A}{dt} = -\gamma\frac{p_A}{V_A}\frac{dV_A}{dt} + \gamma\frac{R}{V_A}\dot{m}_{Ain}T_{in} - \gamma\frac{R}{V_A}\dot{m}_{Aout}T_A + \frac{\gamma-1}{V_A}\lambda_0\sqrt{\left(\frac{p_A T_A}{p_0 T_0}\right)}A_{Aq}(x)(T_{amb}-T_A) \tag{2.29}$$

$$\frac{dT_A}{dt} = \frac{T_A}{V_A}\frac{dV_A}{dt}(1-\gamma) + \dot{m}_{Ain}\frac{RT_A}{V_A p_A}(\gamma T_{in} - T_A) - \dot{m}_{Aout}\frac{RT_A^2}{V_A p_A}(\gamma-1) + \frac{(\gamma-1)T_A}{p_A V_A}\lambda_0\sqrt{\left(\frac{p_A T_A}{p_0 T_0}\right)}A_{Aq}(x)(T_{amb}-T_A) \tag{2.30}$$

$$\frac{dp_B}{dt} = \gamma\frac{p_B}{V_B}\frac{dV_B}{dt} + \gamma\frac{R}{V_B}\dot{m}_{Bin}T_{in} - \gamma\frac{R}{V_B}\dot{m}_{Bout}T_B + \frac{\gamma-1}{V_B}\lambda_0\sqrt{\left(\frac{p_B T_B}{p_0 T_0}\right)}A_{Bq}(x)(T_{amb}-T_B) \tag{2.31}$$

$$\frac{dT_B}{dt} = \frac{T_B}{V_B}\frac{dV_B}{dt}(1-\gamma) + \dot{m}_{Bin}\frac{RT_B}{V_B p_B}(\gamma T_{in} - T_B) - \dot{m}_{Bout}\frac{RT_B^2}{V_B p_B}(\gamma-1) + \frac{(\gamma-1)T_B}{p_B V_B}\lambda_0\sqrt{\left(\frac{p_B T_B}{p_0 T_0}\right)}A_{Bq}(x)(T_{amb}-T_B) \tag{2.32}$$

$$\frac{d^2x}{dt^2} = \frac{p_A A_A - p_B A_B - p_{atm}A_h - F_{fr} - F_{ext}}{m} \tag{2.33}$$

$$\frac{dx}{dt} = \dot{x} \tag{2.34}$$

The mass flow rates entering or leaving the actuator chambers are given by:

$$\dot{m}_{Ain} = \begin{cases} \dot{m}_A & \text{if } \dot{m}_A > 0 \\ 0 & \text{if } \dot{m}_A \leq 0 \end{cases} \quad (2.35) \qquad \dot{m}_{Aout} = \begin{cases} 0 & \text{if } \dot{m}_A \geq 0 \\ \dot{m}_A & \text{if } \dot{m}_A < 0 \end{cases} \quad (2.36)$$

$$\dot{m}_{Bin} = \begin{cases} \dot{m}_B & \text{if } \dot{m}_B > 0 \\ 0 & \text{if } \dot{m}_B \leq 0 \end{cases} \quad (2.37) \qquad \dot{m}_{Bout} = \begin{cases} 0 & \text{if } \dot{m}_B \geq 0 \\ \dot{m}_B & \text{if } \dot{m}_B < 0 \end{cases} \quad (2.38)$$

In the equations above, $\dot{m}_A$ and $\dot{m}_B$ are given by equations 2.19 and 2.20, respectively.

2.5.1 Reduced-Order Non-Linear System Model

The model presented at the end of the previous section is a 6th-order model, whose state variables are pressure and temperature in each chamber along with the piston's velocity and position. It is the model used for the purpose of simulation, but due to its mathematical complexity, it is not suitable for control purposes. Furthermore, it would need a temperature observer, and most temperature sensors do not possess high enough bandwidth to properly track temperature changes such as the ones seen in this application (Falcão Carneiro, 2007). These reasons justify using a simplified model for control purposes, and that simplification is usually done by substituting a differential equation by an algebraic one.

In the context of this work, it is easy to acknowledge that between pressure and temperature, pressure is the most important state variable, with dominant dynamics in terms of piston motion. Common practice is therefore to neglect temperature dynamics, which results in reducing the model from a 6-th order model to a 4-th order one. This is typically done by assuming the thermodynamic process inside the chambers to be polytropic. This means pressure and temperature in a chamber can be related by:

$$T = T_0 \left(\frac{p}{p_0} \right)^{\frac{n-1}{n}} \quad (2.39)$$

where T_0 and p_0 are the equilibrium values of pressure and temperature, and n is the polytropic index, which varies from $n = 1$ to $n = \gamma$.

An extensive study was done by Falcão Carneiro (2007), in which various reduced-order thermodynamic models were analysed. Those models varied in the value of n,

the way in which they considered pressure and temperature variations around the equilibrium values, and in how heat transfer was accounted for.

Of the various models that were tested, the one which showed the best compromise between simplicity and error in pressure prediction, which will also be used in this work, was one that:

- Accounted explicitly for heat transfer between the air and the actuator walls;
- Neglected heat transfer due to mixing of incoming air with air already in the chamber;
- Assumed the thermodynamic process inside the chambers to be polytropic;
- Had $n = 1.35$.

The reduced-order model is then given by:

$$T = T_0 \left(\frac{p}{p_0} \right)^{\frac{n-1}{n}} \tag{2.40}$$

$$\frac{dp}{dt} = -\gamma \frac{p}{V} \frac{dV}{dt} + \gamma \frac{R}{V} T(\dot{m}_{in} - \dot{m}_{out}) + \frac{\gamma - 1}{V} k_0 (T_{amb} - T) \tag{2.41}$$

where k_0 is an average thermal conductance obtained by assuming the heat-transfer area to be constant.

The complete reduced-order system model can then be written as:

$$\frac{dp_A}{dt} = -\gamma \frac{p_A A_A}{V_A} \frac{dx}{dt} + \gamma \frac{R}{V_A} T_A \dot{m}_A + \frac{\gamma - 1}{V_A} k_0 (T_{amb} - T_A) \tag{2.42}$$

$$T_A = T_{A0} \left(\frac{p_A}{p_{A0}} \right)^{\frac{n-1}{n}} \tag{2.43}$$

$$\frac{dp_B}{dt} = \gamma \frac{p_B A_B}{V_B} \frac{dx}{dt} + \gamma \frac{R}{V_B} T_B \dot{m}_B + \frac{\gamma - 1}{V_B} k_0 (T_{amb} - T_B) \tag{2.44}$$

$$T_B = T_{B0} \left(\frac{p_B}{p_{B0}} \right)^{\frac{n-1}{n}} \tag{2.45}$$

$$\frac{d^2x}{dt^2} = \frac{p_A A_A - p_B A_B - p_{atm} A_h - F_{fr} - F_{ext}}{m} \tag{2.46}$$

$$\frac{dx}{dt} = \dot{x} \tag{2.47}$$

2.6 Linearised System Model

As was already discussed in the previous section, the full-order system model was not suitable for control and linearisation purposes, and an order reduction could be achieved without significant loss in accuracy.

This section will deal with the linearisation of the 4-th order model presented at the end of Section 2.5.1. The linearisation will approximate the non-linear system by a linear one around an equilibrium point. This is usually done for controller design purposes, but although that is also true (in a broader sense) in the case of this work, it is not done for the typical reasons, i.e. the linear controller architecture and parameters will ultimately not be defined by analytical tools applied to the linearised model. However, that model will shed light on some important characteristics of the system and provide critical insight to the process of defining the controller architecture.

To present the full model that is to be linearised (Eqs. 2.48 through 2.59), we refer to equations 2.42 through 2.47 and add the mass flow rate equations. Because the control specifications impose an average pressure of 3 bar in the auxiliary chamber (this value will be discussed later on), the pressure in the main chamber is expected to take an average value not far from 3 bar. Since $p_{atm} \approx 1$ bar and $p_s = 7$ bar, we can safely assume choked flow in restrictions R_1 and R_2 - the critical pressure ratios of both restrictions are approximately 0.5 (Falcão Carneiro, 2007). Although for simulation purposes the model accounts for reverse flow through those restrictions, for linearisation purposes we consider only the normal working conditions, i.e. $p_{atm} \leq p_{chamber} \leq p_s$. Furthermore, only viscous friction is considered. This means the friction force is given by $F_{fr} = k_a \dot{x}$, where k_a is a viscous friction coefficient identified by Falcão Carneiro (2007) for the actuator discussed in this work.

$$\frac{dp_A}{dt} = -\gamma \frac{p_A A_A}{V_A} \frac{dx}{dt} + \gamma \frac{R}{V_A} T_A \dot{m}_A + \frac{\gamma - 1}{V_A} k_0 (T_{amb} - T_A) \tag{2.48}$$

$$T_A = T_{A0} \left(\frac{p_A}{p_{A0}} \right)^{\frac{n-1}{n}} \tag{2.49}$$

$$\dot{m}_A = \dot{m}_{A1} - \dot{m}_{A2} \tag{2.50}$$

$$\frac{dp_B}{dt} = -\gamma \frac{p_B A_B}{V_B} \frac{dx}{dt} + \gamma \frac{R}{V_B} T_B \dot{m}_B + \frac{\gamma - 1}{V_B} k_0 (T_{amb} - T_B) \tag{2.51}$$

$$T_B = T_{B0}\left(\frac{p_B}{p_{B0}}\right)^{\frac{n-1}{n}} \tag{2.52}$$

$$\dot{m}_B = \dot{m}_{B1} - \dot{m}_{B2} \tag{2.53}$$

$$\dot{m}_{A1} = C_{A1} p_s \rho_0 \sqrt{\frac{293.15}{T_s}} \tag{2.54}$$

$$\dot{m}_{A2} = C_{A2} p_A \rho_0 \sqrt{\frac{293.15}{T_A}} \tag{2.55}$$

$$\dot{m}_{B1} = C_{B1} p_s \rho_0 \sqrt{\frac{293.15}{T_s}} \tag{2.56}$$

$$\dot{m}_{B2} = C_{B2} p_B \rho_0 \sqrt{\frac{293.15}{T_B}} \tag{2.57}$$

$$\frac{d^2x}{dt^2} = \frac{p_A A_A - p_B A_B - p_{atm} A_h - k_a \dot{x} - F_{ext}}{m} \tag{2.58}$$

$$\frac{dx}{dt} = \dot{x} \tag{2.59}$$

2.6.1 Taylor-Series Expansion

The dynamics of the 4th-order non-linear model can be succinctly expressed as:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{2.60}$$

where $\dot{\mathbf{x}}$, $\mathbf{x}$ and $\mathbf{u}$ are:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{p_A} \\ \dot{p_B} \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x \\ \dot{x} \\ p_A \\ p_B \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} u_A \\ u_B \end{bmatrix}$$

To perform the linearisation, we resort to a Taylor-series expansion around the equilibrium point $\mathbf{x}_0$, using solely the first-term of the series. Equation 2.60 can then be approximated, in the vicinity of $\mathbf{x}_0$, as:

$$\dot{\mathbf{x}} \approx \left.\frac{\partial f(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right|_0 \delta\mathbf{x} + \left.\frac{\partial f(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\right|_0 \delta\mathbf{u} \tag{2.61}$$

where:

$$\delta\mathbf{x} = \mathbf{x} - \mathbf{x}_0 \tag{2.62}$$

$$\delta\mathbf{u} = \mathbf{u} - \mathbf{u}_0 \tag{2.63}$$

We can then rewrite equation 2.61 as:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{p_A} \\ \dot{p_B} \end{bmatrix} = \begin{bmatrix} \left.\frac{\partial \dot{x}}{\partial x}\right|_0 & \left.\frac{\partial \dot{x}}{\partial \dot{x}}\right|_0 & \left.\frac{\partial \dot{x}}{\partial p_A}\right|_0 & \left.\frac{\partial \dot{x}}{\partial p_B}\right|_0 \\ \left.\frac{\partial \ddot{x}}{\partial x}\right|_0 & \left.\frac{\partial \ddot{x}}{\partial \dot{x}}\right|_0 & \left.\frac{\partial \ddot{x}}{\partial p_A}\right|_0 & \left.\frac{\partial \ddot{x}}{\partial p_B}\right|_0 \\ \left.\frac{\partial \dot{p_A}}{\partial x}\right|_0 & \left.\frac{\partial \dot{p_A}}{\partial \dot{x}}\right|_0 & \left.\frac{\partial \dot{p_A}}{\partial p_A}\right|_0 & \left.\frac{\partial \dot{p_A}}{\partial p_B}\right|_0 \\ \left.\frac{\partial \dot{p_B}}{\partial x}\right|_0 & \left.\frac{\partial \dot{p_B}}{\partial \dot{x}}\right|_0 & \left.\frac{\partial \dot{p_B}}{\partial p_A}\right|_0 & \left.\frac{\partial \dot{p_B}}{\partial p_B}\right|_0 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta \dot{x} \\ \delta p_A \\ \delta p_B \end{bmatrix} + \begin{bmatrix} \left.\frac{\partial \dot{x}}{\partial u_A}\right|_0 & \left.\frac{\partial \dot{x}}{\partial u_B}\right|_0 \\ \left.\frac{\partial \ddot{x}}{\partial u_A}\right|_0 & \left.\frac{\partial \ddot{x}}{\partial u_B}\right|_0 \\ \left.\frac{\partial \dot{p_A}}{\partial u_A}\right|_0 & \left.\frac{\partial \dot{p_A}}{\partial u_B}\right|_0 \\ \left.\frac{\partial \dot{p_B}}{\partial u_A}\right|_0 & \left.\frac{\partial \dot{p_B}}{\partial u_B}\right|_0 \end{bmatrix} \begin{bmatrix} \delta u_A \\ \delta u_B \end{bmatrix} + \begin{bmatrix} \left.\frac{\partial \dot{x}}{\partial F_{ext}}\right|_0 \\ \left.\frac{\partial \ddot{x}}{\partial F_{ext}}\right|_0 \\ \left.\frac{\partial \dot{p_A}}{\partial F_{ext}}\right|_0 \\ \left.\frac{\partial \dot{p_B}}{\partial F_{ext}}\right|_0 \end{bmatrix} \delta F_{ext} \tag{2.64}$$

2.6.2 Equilibrium Conditions

The equilibrium state $\mathbf{x}_0$ must now be defined. We consider the system to be in its equilibrium state if:

- $\dot{\mathbf{x}} = 0$, which trivially results in:

$$\frac{dx}{dt} = 0 \tag{2.65}$$

- $\ddot{\mathbf{x}} = 0$, which, considering $F_{ext} = 0$, results in:

$$p_A A_A - p_B A_B - p_{atm} A_h - k_a \dot{x} = 0 \tag{2.66}$$

For $\dot{\mathbf{x}} = 0$, the equation above reduces to:

$$p_A A_A - p_B A_B - p_{atm} A_h = 0 \tag{2.67}$$

- $\dot{p_A} = 0$, which results (for $\dot{x} = 0$) in:

$$0 = \gamma \frac{R}{V_A} T_A \dot{m}_A + \frac{\gamma - 1}{V_A} k_0 (T_{amb} - T_A) \tag{2.68}$$

- $\dot{p_B} = 0$, which results (for $\dot{x} = 0$) in:

$$0 = \gamma \frac{R}{V_B} T_B \dot{m}_B + \frac{\gamma - 1}{V_B} k_0 (T_{amb} - T_B) \tag{2.69}$$

Concerning this equilibrium condition, it should be noted that according to the polytropic law, if pressure is constant ($\dot{p}_{A/B} = 0$), so is temperature. Since there is only one way the air can enter or exit a chamber, it follows that if pressure, temperature and volume are constant, than there must be no air entering or leaving that chamber ($\dot{m}_{A/B}\Big|_0 = 0$). All the aforementioned conditions imply that there also should not be any heat transfer between the chambers and the environment, and so the equilibrium temperatures must be equal to the ambient temperature: $T_{A0} = T_{B0} = T_{amb}$.

The next step is to define the equilibrium position x_0 and to find the equilibrium values of the pressures (p_{A0} and p_{B0}) and control actions (u_{A0} and u_{B0}).

The equilibrium position is defined as $x_0 = 0$ for the sake of simplicity, since there are no formal restrictions on the choosing of this position. This allows us to obtain the chamber volumes at equilibrium:

$$V_{A0} = A_A \frac{l}{2} + V_d \tag{2.70}$$

$$V_{B0} = A_B \frac{l}{2} + V_d \tag{2.71}$$

Regarding the equilibrium pressures, we already have a control specification which states that the controller of the auxiliary chamber acts as a regulator, maintaining the pressure in that chamber at 3 bar. This means there will be different equilibrium pressures and control actions for forward and backward movements.

The procedure used for finding the equilibrium values is as follows:

- In the simulation environment, using the full-order model, force a constant value of $p_{aux} = 3\text{x}10^5$ Pa in the auxiliary chamber.
- Find the range of control action (u) values for which the system stays at rest, and take the median value. This is the equilibrium control action for the main chamber controller. The pressure value in the other chamber is the equilibrium value for said chamber.
- Remove the restriction of $p_{aux} = 3\text{x}10^5$ Pa and, for the main chamber control action found in the previous step, adjust the auxiliary chamber control action until the system is shown to stay at rest with a steady-state pressure in the auxiliary chamber of $p_{aux} = 3\text{x}10^5$ Pa.

The system equilibrium values are shown, for each movement, in tables 2.4 and 2.5.

Table 2.4 Equilibrium Values for Forward Movement

$x\ (m)$	$\dot{x}\ (\mathrm{m\,s^{-1}})$	$T\ (K)$	$V\ (\mathrm{m^3})$	$u\ (V)$	p (Pa)
$x_0 = 0$	$\dot{x}_0 = 0$	$T_{A0} = 293.15$	$V_{A0} = 2.124\text{x}10^{-4}$	$u_{A0} = 4.4$	$p_{A0} = 2.8\text{x}10^{5}$
		$T_{B0} = 293.15$	$V_{B0} = 1.897\text{x}10^{-4}$	$u_{B0} = 4.428$	$p_{B0} = 3\text{x}10^{5}$

Table 2.5 Equilibrium Values for Backward Movement

$x\ (m)$	$\dot{x}\ (\mathrm{m\,s^{-1}})$	$T\ (K)$	$V\ (\mathrm{m^3})$	$u\ (V)$	p (Pa)
$x_0 = 0$	$\dot{x}_0 = 0$	$T_{A0} = 293.15$	$V_{A0} = 2.124\text{x}10^{-4}$	$u_{A0} = 4.428$	$p_{A0} = 3\text{x}10^{5}$
		$T_{B0} = 293.15$	$V_{B0} = 1.897\text{x}10^{-4}$	$u_{B0} = 4.47$	$p_{B0} = 3.36\text{x}10^{5}$

2.6.3 Coefficients of the Linearised Model

We will now proceed to find the coefficients of the linearised model, i.e. the partial derivatives in equation 2.64.

For $\frac{dx}{dt} = \dot{x}$, we have:

$$\left.\frac{\partial \dot{x}}{\partial \dot{x}}\right|_0 = 1 \tag{2.72}$$

$$\left.\frac{\partial \dot{x}}{\partial x}\right|_0 = \left.\frac{\partial \dot{x}}{\partial p_A}\right|_0 = \left.\frac{\partial \dot{x}}{\partial p_B}\right|_0 = \left.\frac{\partial \dot{x}}{\partial u_A}\right|_0 = \left.\frac{\partial \dot{x}}{\partial u_B}\right|_0 = \left.\frac{\partial \dot{x}}{\partial F_{ext}}\right|_0 = 0 \tag{2.73}$$

For $\frac{d^2x}{dt^2} = \frac{p_A A_A - p_B A_B - p_{atm} A_h - k_a \dot{x} - F_{ext}}{m}$, we have:

$$\left.\frac{\partial \ddot{x}}{\partial x}\right|_0 = 0 \tag{2.74}$$

$$\left.\frac{\partial \ddot{x}}{\partial \dot{x}}\right|_0 = -\frac{k_a}{m} \tag{2.75}$$

$$\left.\frac{\partial \ddot{x}}{\partial p_A}\right|_0 = \frac{A_A}{m} \tag{2.76}$$

$$\left.\frac{\partial \ddot{x}}{\partial p_B}\right|_0 = \frac{A_B}{m} \tag{2.77}$$

$$\left.\frac{\partial \ddot{x}}{\partial u_A}\right|_0 = \left.\frac{\partial \ddot{x}}{\partial u_B}\right|_0 = 0 \tag{2.78}$$

$$\left.\frac{\partial \ddot{x}}{\partial F_{ext}}\right|_0 = -\frac{1}{m} \tag{2.79}$$

For $\frac{dp_A}{dt} = -\gamma\frac{p_A A_A}{V_A}\frac{dx}{dt} + \gamma\frac{R}{V_A}T_A\dot{m}_A + \frac{\gamma-1}{V_A}k_0(T_{amb} - T_A)$, we have:

$$\left.\frac{\partial\dot{p}_A}{\partial x}\right|_0 = \left.\frac{\partial\dot{p}_A}{\partial p_B}\right|_0 = \left.\frac{\partial\dot{p}_A}{\partial u_B}\right|_0 = \left.\frac{\partial\dot{p}_A}{\partial F_{ext}}\right|_0 = 0 \tag{2.80}$$

$$\left.\frac{\partial\dot{p}_A}{\partial\dot{x}}\right|_0 = -\frac{\gamma p_{A0} A_A}{V_{A0}} = \psi_A \tag{2.81}$$

$$\left.\frac{\partial\dot{p}_A}{\partial p_A}\right|_0 = \left(\frac{\gamma R T_A}{V_A}G_{pA}\right)\Big|_0 - \left(\frac{(\gamma-1)k_0}{V_A}\frac{\partial T_A}{\partial p_A}\right)\Big|_0 \tag{2.82}$$

where the partial derivative of temperature relative to pressure is given by

$$\frac{\partial T_A}{\partial p_A} = \frac{T_{A0}}{p_{A0}^{\frac{n-1}{n}}}\frac{n-1}{n}p_A^{-1/n} \implies \left.\frac{\partial T_A}{\partial p_A}\right|_0 = \frac{n-1}{n}\frac{T_{A0}}{p_{A0}} \tag{2.83}$$

and G_{pA} is the mass flow/pressure coefficient of servo-valve A, given by (Falcão Carneiro, 2007):

$$G_{pA} = \frac{\partial\dot{m}_A}{\partial p_A} = -\frac{\partial\dot{m}_A}{\partial u_A}\Big/\frac{\partial p_A}{\partial u_A} \tag{2.84}$$

where $\frac{\partial\dot{m}_A}{\partial u_A} = G_{uA}$ is the flow-rate gain and $\frac{\partial p_A}{\partial u_A}$ is the pressure gain. Both these coefficients will be further discussed later on. Finally, we have:

$$\left.\frac{\partial\dot{p}_A}{\partial u_A}\right|_0 = \left(-\frac{\gamma R T_A}{V_A}G_{uA}\right)\Big|_0 \tag{2.85}$$

For $\frac{dp_B}{dt} = -\gamma\frac{p_B A_B}{V_B}\frac{dx}{dt} + \gamma\frac{R}{V_B}T_B\dot{m}_B + \frac{\gamma-1}{V_B}k_0(T_{amb} - T_B)$, we follow the same procedure:

$$\left.\frac{\partial \dot{p}_B}{\partial x}\right|_0 = \left.\frac{\partial \dot{p}_B}{\partial p_A}\right|_0 = \left.\frac{\partial \dot{p}_B}{\partial u_A}\right|_0 = \left.\frac{\partial \dot{p}_B}{\partial F_{ext}}\right|_0 = 0 \tag{2.86}$$

$$\left.\frac{\partial \dot{p}_B}{\partial \dot{x}}\right|_0 = -\frac{\gamma p_{B0} A_B}{V_{B0}} = \psi_B \tag{2.87}$$

$$\left.\frac{\partial \dot{p}_B}{\partial p_B}\right|_0 = \left.\left(\frac{\gamma R T_B}{V_B}G_{pB}\right)\right|_0 - \left.\left(\frac{(\gamma-1)k_0}{V_B}\frac{\partial T_B}{\partial p_B}\right)\right|_0 \tag{2.88}$$

$$\left.\frac{\partial T_B}{\partial p_B}\right|_0 = \frac{n-1}{n}\frac{T_{B0}}{p_{B0}} \tag{2.89}$$

$$\left.\frac{\partial \dot{p}_B}{\partial u_B}\right|_0 = \left.\left(-\frac{\gamma R T_B}{V_B}G_{uB}\right)\right|_0 \tag{2.90}$$

We can now proceed to find the flow-rate gains G_{pA}, G_{pB} and pressure gains $\frac{\partial p_A}{\partial u_A}$, $\frac{\partial p_B}{\partial u_B}$.

The pressure gains were obtained through a relatively simple procedure:

- In the simulation environment, for the equilibrium conditions, change u_A from u_{A0} to $u_A = u_{A0} - 0.1$, while forcing $\dot{x} = 0$ and $\ddot{x} = 0$. Register the new steady-state value of p_A. Repeat the procedure but for $u_A = u_{A0} + 0.1$.
- Obtain the pressure gain of servo-valve A by finding the slope of the line connecting the steady-state pressures $p_{A0}|_{u_{A0}-0.1}$ and $p_{A0}|_{u_{A0}+0.1}$

$$\left.\frac{\partial p_A}{\partial u_A}\right|_0 = \frac{p_{A0}|_{u_{A0}+0.1} - p_{A0}|_{u_{A0}-0.1}}{0.2} \tag{2.91}$$

- Repeat the procedure for servo-valve B, which results in:

$$\left.\frac{\partial p_B}{\partial u_B}\right|_0 = \frac{p_{B0}|_{u_{B0}+0.1} - p_{B0}|_{u_{B0}-0.1}}{0.2} \tag{2.92}$$

In order to find G_{pA} and G_{pB}, we must first find the flow-rate gains $\frac{\partial \dot{m}_A}{\partial u_A}$ and $\frac{\partial \dot{m}_B}{\partial u_B}$. For that, we resort to a similar procedure:

- In the simulation environment, for the equilibrium conditions, change u_A from u_{A0} to $u_A = u_{A0} - 0.1$, while forcing $\dot{x} = 0$ and $\ddot{x} = 0$. Register the steady state values of $\dot{m}_{Ain}$ and $\dot{m}_{Aout}$. Repeat the procedure but for $u_A = u_{A0} + 0.1$.
- Obtain $\frac{\partial \dot{m}_{Ain}}{\partial \dot{u}_A}$:

$$\left.\frac{\partial \dot{m}_{Ain}}{\partial \dot{u}_A}\right|_0 = \frac{\dot{m}_{Ain}|_{u_{A0}+0.1} - \dot{m}_{Ain}|_{u_{A0}-0.1}}{0.2} \tag{2.93}$$

- Obtain $\frac{\partial \dot{m}_{Aout}}{\partial \dot{u}_A}$:

$$\left.\frac{\partial \dot{m}_{Aout}}{\partial \dot{u}_A}\right|_0 = \frac{\dot{m}_{Aout}|_{u_{A0}+0.1} - \dot{m}_{Aout}|_{u_{A0}-0.1}}{0.2} \tag{2.94}$$

- Obtain the flow-rate gain of servo-valve A:

$$G_{uA} = \left.\frac{\partial \dot{m}_A}{\partial \dot{u}_A}\right|_0 = \left.\frac{\partial \dot{m}_{Ain}}{\partial \dot{u}_A}\right|_0 - \left.\frac{\partial \dot{m}_{Aout}}{\partial \dot{u}_A}\right|_0 \tag{2.95}$$

- Repeat procedure for servo-valve B. This results in:

$$\left.\frac{\partial \dot{m}_{Bin}}{\partial \dot{u}_B}\right|_0 = \frac{\dot{m}_{Bin}|_{u_{B0}+0.1} - \dot{m}_{Bin}|_{u_{B0}-0.1}}{0.2} \tag{2.96}$$

$$\left.\frac{\partial \dot{m}_{Bout}}{\partial \dot{u}_B}\right|_0 = \frac{\dot{m}_{Bout}|_{u_{B0}+0.1} - \dot{m}_{Bout}|_{u_{B0}-0.1}}{0.2} \tag{2.97}$$

$$G_{uB} = \left.\frac{\partial \dot{m}_B}{\partial \dot{u}_B}\right|_0 = \left.\frac{\partial \dot{m}_{Bin}}{\partial \dot{u}_B}\right|_0 - \left.\frac{\partial \dot{m}_{Bout}}{\partial \dot{u}_B}\right|_0 \tag{2.98}$$

We can then compute G_{pA} and G_{pB}:

$$G_{pA} = \frac{\partial \dot{m}_A}{\partial p_A} = -\frac{\partial \dot{m}_A}{\partial u_A} \Big/ \frac{\partial p_A}{\partial u_A} \tag{2.99}$$

$$G_{pB} = \frac{\partial \dot{m}_B}{\partial p_B} = -\frac{\partial \dot{m}_B}{\partial u_B} \Big/ \frac{\partial p_B}{\partial u_B} \tag{2.100}$$

The values herein obtained were then corroborated by a MATLAB® script that calculated those coefficients based on the equilibrium conditions, the mass flow-rate equations and experimental data of the servo-valves (Falcão Carneiro, 2007). Table 2.6 presents the values obtained in simulation and with the script.

Table 2.6 Coefficients of the linearised model

	G_{uA}	G_{uB}	G_{pA}	G_{pB}
Simulation	$1.1\text{x}10^{-3}$	$1\text{x}10^{-3}$	$-1.158\text{x}10^{-9}$	$-1.091\text{x}10^{-9}$
Script	$1.2\text{x}10^{-3}$	$1\text{x}10^{-3}$	$-1.256\text{x}10^{-9}$	$-1.143\text{x}10^{-9}$

We can see that the results are in agreement with each other. Because they were given by the model itself, the values given in the simulation tests are used. In the next page, Table 2.7 features the numeric values of all remaining system parameters. Recall that the parameters of the LuGre model were already shown in Table 2.3, and the equilibrium values for the control actions, pressures and temperatures are in Tables 2.2 and 2.3.

Table 2.7 Model Parameters

Parameter	Value	Units
A_A	$8.0425\text{x}10^{-4}$	m^2
A_B	$6.9115\text{x}10^{-4}$	m^2
A_h	$1.131\text{x}10^{-4}$	m^2
λ_0	22.75	$\mathrm{W\,m^{-2}\,K^{-1}}$
G_{pA}	$-1.158\text{x}10^{-9}$	$\mathrm{kg\,Pa^{-1}\,s^{-1}}$
G_{pB}	$-1.091\text{x}10^{-9}$	$\mathrm{kg\,Pa^{-1}\,s^{-1}}$
G_{uA}	$1.1\text{x}10^{-3}$	$\mathrm{kg\,V^{-1}\,s^{-1}}$
G_{uB}	$1\text{x}10^{-3}$	$\mathrm{kg\,V^{-1}\,s^{-1}}$
k_0	0.1	$\mathrm{W\,K^{-1}}$
k_a	57.81	$\mathrm{N\,s\,m^{-1}}$
ρ_0	1.2926	$\mathrm{kg\,m^{-3}}$
n	1.35	-
p_{atm}	1	bar
p_s	7	bar
R	287.1	$\mathrm{J\,kg^{-1}\,K^{-1}}$
T_{amb}	293.15	K
T_s	293.15	K

2.6.4 4th-Order Linearised Model

Having found all the coefficients of equation 2.64, we can now rewrite it in a state-space representation to present the 4-th order linearised system model. Note that since $x_0 = 0$, $\dot{x}_0 = 0$ and $F_{ext}|_0 = 0$, we have $\delta x = x$, $\delta \dot{x} = \dot{x}$ and $\delta F_{ext} = F_{ext}$. Let us first recall that a state-space representation for a continuous, linear time-invariant system takes the form of:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{2.101}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{2.102}$$

where $\mathbf{x}$ is the state vector, $\mathbf{y}$ is the output vector and $\mathbf{u}$ is the input (or control) vector. $\mathbf{A}$ is the state matrix, $\mathbf{B}$ is the input matrix, $\mathbf{C}$ is the output matrix and $\mathbf{D}$ is the direct feedthrough matrix.

There are 4 state variables: $x, \dot{x}, p_A$ and p_B. The state matrix $\mathbf{A}$ and input matrix $\mathbf{B}$ will be given by:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{k_a}{m} & \frac{A_A}{m} & -\frac{A_B}{m} \\ 0 & \psi_A & -\frac{1}{\tau_A} & 0 \\ 0 & \psi_B & 0 & -\frac{1}{\tau_B} \end{bmatrix} \tag{2.103}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{\gamma R T_{A0}}{V_{A0}} G_{uA}|_0 & 0 \\ 0 & \frac{\gamma R T_{B0}}{V_{B0}} G_{uB}|_0 \end{bmatrix} \tag{2.104}$$

Since we are interested in studying how velocity and pressure in the auxiliary chamber relate, and seeing as there are no direct feedthrough terms that bypass the system dynamics, the output matrix $\mathbf{C}$ and the feedthrough matrix $\mathbf{D}$ are given by:

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.105}$$

$$\mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.106}$$

Assuming no external forces acting on the system, its state-space representation is then given by:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{p}_A \\ \dot{p}_B \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{k_a}{m} & \frac{A_A}{m} & -\frac{A_B}{m} \\ 0 & \psi_A & -\frac{1}{\tau_A} & 0 \\ 0 & \psi_B & 0 & -\frac{1}{\tau_B} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x \\ \dot{x} \\ \delta p_A \\ \delta p_B \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{\gamma R T_{A0}}{V_{A0}} G_{uA}|_0 & 0 \\ 0 & \frac{\gamma R T_{B0}}{V_{B0}} G_{uB}|_0 \end{bmatrix}}_{\mathbf{B}} \begin{bmatrix} \delta u_A \\ \delta u_B \end{bmatrix} \tag{2.107}$$

$$\begin{bmatrix} y1 \\ y2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} x \\ \dot{x} \\ \delta p_A \\ \delta p_B \end{bmatrix} \tag{2.108}$$

τ_A and τ_B represent the time-constants of chamber A and B, respectively. They are given by:

$$\tau_A = -\frac{1}{\frac{\partial \dot{p}_A}{\partial p_A}\Big|_0} = -\frac{1}{\left(\frac{\gamma R T_A}{V_A} G_{pA}\right)\Big|_0 - \frac{(\gamma-1)k_0}{V_{A0}} \frac{n-1}{n} \frac{T_{A0}}{p_{A0}}} \tag{2.109}$$

$$\tau_B = -\frac{1}{\frac{\partial \dot{p}_B}{\partial p_B}\Big|_0} = -\frac{1}{\left(\frac{\gamma R T_B}{V_B} G_{pB}\right)\Big|_0 - \frac{(\gamma-1)k_0}{V_{B0}} \frac{n-1}{n} \frac{T_{B0}}{p_{B0}}} \tag{2.110}$$

The linear system described by equations 2.107 and 2.108 can also be visualised in the form of a simulation diagram:

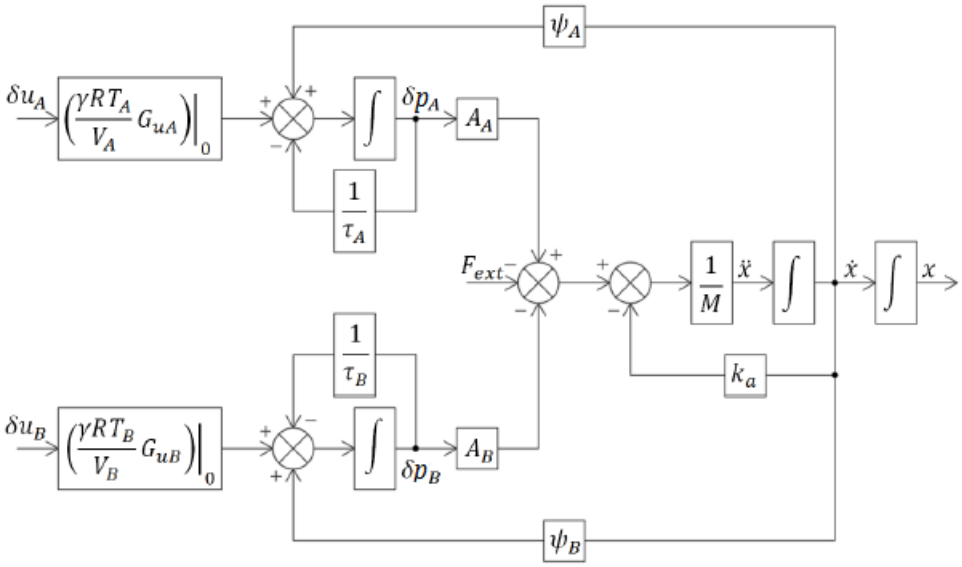

Fig. 2.9 Simulation diagram representation of the 4th-order linear system model (Pinto, 2017)

Before advancing, it would be interesting to see how the time constants of the chambers vary with their volume. It is trivial to understand that the time constant of a chamber will rise as the volume increases and fall with decreasing volume, but this will give us an idea of how different the time constants really are when the piston is close to one of the actuator ends. This is relevant to this work because, having independent control of the two servo-valves, the "balance" between the controllers is affected if one chamber's dynamics are much faster than the other's. Consider a situation where, for example, the piston advances in the opposite direction of an end that is very close. The motion controller will drop the pressure in the auxiliary chamber to make the necessary pressure differential for acceleration, and the pressure controller will try to

bring that pressure back up to 3 bar. But because of the small volume, the pressure in the main chamber reacts very quickly to changes in the pressure-loop control action, while in the other chamber the pressure takes a longer time to react to motion-loop control action changes. Note that this information is not directly used in the control design process, but it provides some insight into the overall behaviour of the system.

With the coefficients of Table 2.6, we can plot the time constants (equations 2.109 and 2.110) as functions of the piston position (and therefore chamber volumes).

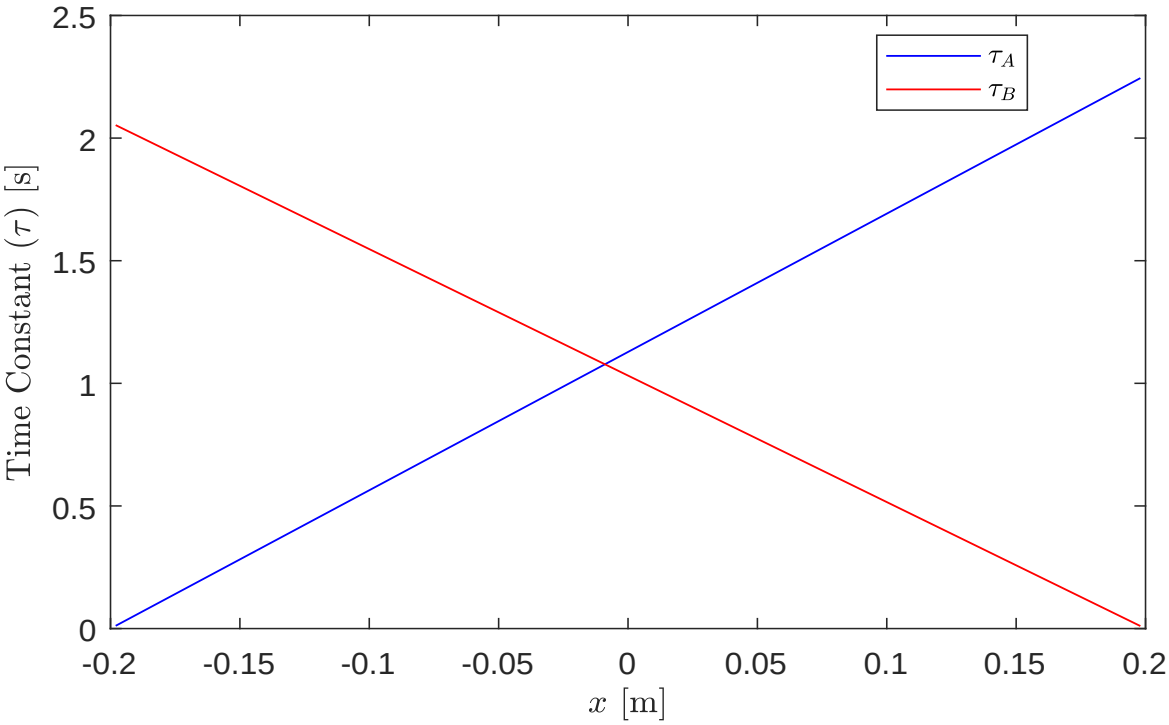

Fig. 2.10 Time constants τ_A and τ_B as a function of piston position.

It must be noted that these are merely approximations of the time-constants, by-products of the linearisation of the system. Nevertheless, we can see that there are very large differences near the ends of the actuator - from a few tens of milliseconds to more than 2 seconds.

With the 4th-order linear system model properly described and its coefficients found, it is the right time to break away from the literature and try to establish a conceptual link between what we can learn from certain representations of the linear model and the control-wise objectives of this work.

As was previously discussed, the general configuration of the controller will contemplate a pressure regulator connected to the the main chamber (although regulating pressure in the auxiliary chamber) and a cascaded motion controller in the auxiliary chamber, where a proportional position controller provides the input for a velocity controller. In other words, the "heavy work" is done by the velocity and pressure controllers, and then the position loop is closed around the velocity controller, providing it with an input proportional to the positioning error. We thus have, before closing the position loop, two control variables: pressure in the auxiliary chamber and piston velocity. This means the controlled system is a multi-input, multi-output (MIMO) system, which of course differs from a single-input, single-output (SISO) system.

It is important to introduce this MIMO vs SISO discussion because many SISO techniques used to study the stability of a system (root-locus, for example) don't translate well to MIMO systems. This happens because in such systems, closing a loop affects not only the output that we wish to control, it affects all outputs, i.e. in closed loop, the dynamics that govern the behaviour of each output variable are influenced (in varying degrees) by all control inputs. We therefore cannot, generally, independently control the various output variables. This cross-coupling of outputs becomes clearer in a transfer function representation of the system, which we will present in the following section. What we learn can then be used further along in the control-design phase.

Transfer Function Representation

The state-space representation can now be transformed into a transfer-function representation. It should be noted that in a MIMO system, the dynamics are of course not described by a sole transfer function, but by a matrix of them. That transfer function matrix is composed of the individual transfer functions that relate each input to each output.

The transfer-function matrix $\mathbf{G}$ is of the form

$$\mathbf{G}(s) = \frac{\mathbf{Y}(s)}{\mathbf{U}(s)} \tag{2.111}$$

and it can be obtained from the state-space representation through the following equation (Ogata, 2001):

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \tag{2.112}$$

where s is the Laplace variable and I is the identity matrix.

In this case, equation 2.112 gives us:

$$\begin{bmatrix} X \\ \dot{X} \\ \delta p_A \\ \delta p_B \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \\ G_{31} & G_{32} \\ G_{41} & G_{42} \end{bmatrix} \cdot \begin{bmatrix} \delta U_a \\ \delta U_b \end{bmatrix} \tag{2.113}$$

The open-loop transfer functions of interest are:

$$\frac{\dot{X}(s)}{\delta U_b(s)} = G_{22}(s) = \frac{\frac{A_B}{V_B}\frac{\gamma R T_{B0}}{m} G_{uB}(-\frac{1}{\tau_A} - s)}{Den} \tag{2.114}$$

$$\frac{\delta p_B(s)}{\delta U_a(s)} = G_{41}(s) = \frac{\psi_B \frac{A_A}{V_A}\frac{\gamma R T_{A0}}{m} G_{uA}}{Den} \tag{2.115}$$

with:

$$Den = s^3 + \left(\frac{k_a}{m} + \frac{1}{\tau_A} + \frac{1}{\tau_B}\right)s^2 + \left(\frac{k_a}{m}(\frac{1}{\tau_A} + \frac{1}{\tau_B}) + \frac{1}{\tau_A\tau_B} - \frac{\psi_A A_A}{m} + \frac{\psi_B A_B}{m} - \frac{A_B}{V_B}\frac{\gamma R T_{B0}}{m} G_{uB}\right)s - \frac{\psi_A A_A}{\tau_B m} + \frac{\psi_B A_B}{\tau_A m} + \frac{K_a}{m\tau_A\tau_B} \tag{2.116}$$

We can get some information from these transfer functions: note that G_{41} has three poles and no zeros - this means that, in closed-loop, there is a limit to the proportional gain after which the system becomes unstable. G_{22} represents a positive-feedback system, but it has relative order 2 (three poles and one zero), so theoretically, as long as the proportional gain is negative, the closed-loop poles of the system will remain in the left-half s plane.

It is important to note that in practice, $\dot{x}$ will be estimated by a Kalman filter (Section 3.3.3), since it cannot be measured directly - that estimate can then be used by the controllers. Since the Kalman filter also has an inherent dynamic response, the

use of its estimates in a feedback loop will alter the closed-loop transfer functions, and thus the assumptions made in the previous paragraph might not prove true in practice.

Having made a brief qualitative analysis of the transfer functions of interest, let us now return to the coupling discussion. We can take the transfer functions relating to our variables of interest, $\dot{x}$ and p_B (in a forward movement), and represent them in a block diagram:

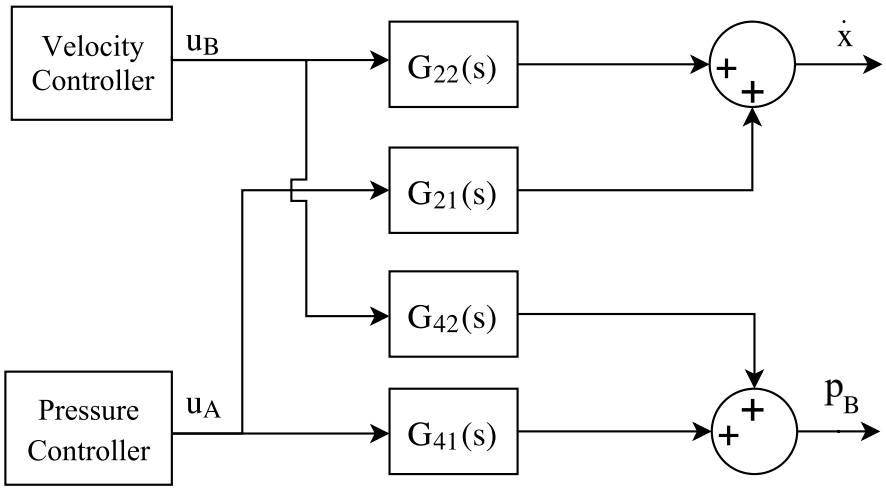

Fig. 2.11 Open-loop block diagram of pressure and velocity

It becomes clear from Fig. 2.11 how the pressure control-loop influences the motion control-loop and vice-versa. This will of course be taken into account in the control design phase. The following section provides some more practical insight into the coupling between velocity and pressure and evaluates how some physical properties of the system "naturally" decouple those variables in certain conditions.

2.6.5 Coupling of Pressure and Velocity - A Brief Analysis

As we saw at the end of the previous section, the dynamics of pressure and velocity are coupled. Despite this, some characteristics of the system tend to decouple those dynamics in the right conditions.

The piston velocity, as we know, is limited by the volumetric flow rate of the air leaving the auxiliary chamber through the exhaust port. In normal conditions, that flow rate is a function of (among other variables) upstream and downstream pressure. Since the downstream pressure of the exhaust is constant and equal to p_{atm}, then we can see that the volumetric flow rate, for a given, constant control action u_B (forward movement), is still influenced by the upstream pressure p_B, and consequently by u_A.

But this is not always the case - if the flow is choked, i.e. the volumetric flow rate is independent of the pressure ratio, then for a constant u_B, the steady-state piston velocity remains essentially unaffected by upstream pressure fluctuations, and is thus shielded from the influence of u_A. It is relatively easy to achieve choked flow through the exhaust port, since the theoretical critical pressure ratio is close to 0.5 (Falcão Carneiro, 2007), so an upstream pressure of 2 bar is enough to develop choked flow. Keeping in mind that the pressure in the auxiliary chamber will be regulated to 3 bar, it is safe to assume that, even considering ± 0.5 bar pressure variations, the flow will remain choked.

That being said, one might suggest that the coupling problem is actually no problem at all, since the condition of choked flow seems to shield piston velocity from pressure variations. Unfortunately, that isn't so. Seeing as there are always leaks, there's also flow through restriction R_1, between the chamber and the pressure port.

If we recall equation 2.19, we see that the net flow through the working port is composed of the flows through both restrictions, so this leak flow affects the overall velocity dynamics because, as we will see, it makes the piston velocity sensible to pressure variations in the chamber.

It is therefore interesting to investigate how velocity is affected by pressure in the auxiliary chamber. It is logical to expect that the effect will be more noticeable at low velocities, where the leak flow through R_1 is significant relative to the "main" flow through R_2.

We begin by referring to the schematic in Fig. 2.12.

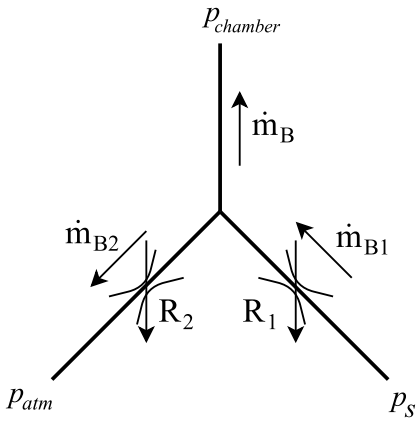

Fig. 2.12 Schematic representation of the servo-valve restrictions

We henceforth assume that the discussion refers to a forward movement, where servo-valve B is responsible for the velocity control. We recall equation 2.20 and rewrite $\dot{m}_B$ as:

$$\dot{m}_B = \rho_B Q_B \tag{2.117}$$

Using the ideal gas law and conservation of mass principle, we rewrite ρ_B and Q_B:

$$\dot{m}_B = \frac{p_B}{RT_B}\dot{x}A_B \tag{2.118}$$

So since

$$\dot{m}_B = \dot{m}_{B1} - \dot{m}_{B2} \tag{2.119}$$

and assuming choked flow for both restrictions, we have:

$$\frac{p_B}{RT_B}\dot{x}A_B = C_{B1}p_s\rho_0\sqrt{\frac{293.15}{T_s}} - C_{B2}p_B\rho_0\sqrt{\frac{293.15}{T_B}} \tag{2.120}$$

which can be rewritten as:

$$\dot{x} = \frac{RT_B}{A_B}C_{B1}\frac{p_s}{p_B}\rho_0\sqrt{\frac{293.15}{T_s}} - \frac{RT_B}{A_B}C_{B2}\rho_0\sqrt{\frac{293.15}{T_B}} \tag{2.121}$$

Note that in equation 2.121, if we were to remove the first term (relative to $\dot{m}_{B1}$), the dependence of p_B would disappear:

$$\dot{x} = -\frac{RT_B}{A_B} C_{B2} \rho_0 \sqrt{\frac{293.15}{T_B}} \tag{2.122}$$

Going back to equation 2.121, it is interesting then to take its derivative relative to p_B, and see its evolution for different steady-state velocity values. Equation 2.123 represents that partial derivative.

$$\frac{\partial \dot{x}}{\partial p_B} = -\frac{RT_B}{A_B} C_{B1} p_s \rho_0 \sqrt{\frac{293.15}{T_s}} \frac{p_s}{p_B^2} \tag{2.123}$$

In order to find the values of this partial derivative, we must define a set of steady-state velocity values to test, and find the sonic conductances corresponding to the steady-state control actions u_B.

The procedure for this analysis was as follows:

- Using an initial version of the controller to achieve the desired steady-state velocities, take the steady-state values of u_B for each of those velocities, along with the mass flow rates $\dot{m}_{B1}$ and $\dot{m}_{B2}$ so that the leak flow can be quantified.
- Take the sonic conductance values C_{B1} corresponding to each steady-state control action u_B.
- Using the values found in the previous steps, calculate $\left.\frac{\partial \dot{x}}{\partial p_B}\right|_{p_{B0}}$ for each steady-state velocity using Eq. 2.123.
- Finally, knowing the values of $\left.\frac{\partial \dot{x}}{\partial p_B}\right|_{p_{B0}}$, calculate the percentage velocity variation for a pressure variation (p_B) of 1 bar, for each value of the steady-state velocity.

Table 2.8 shows the results obtained in the various steps above.

Table 2.8 Test results for influence of $\dot{x}$ on $\left.\frac{\partial \dot{x}}{\partial p_B}\right|_{p_{B0}}$

$\dot{x}_{ss}$ ($\mathrm{m\,s^{-1}}$)	$\frac{\dot{m}_{B1}}{\dot{m}_{B2}}$ (%)	u_{Bss} (V)	C_{r1}x10^{10}	$\left\lvert\frac{\partial \dot{x}}{\partial p_B}\right\rvert_{p_{B0}}$ x10^{7} ($\mathrm{m\,s^{-1}\,Pa^{-1}}$)
0.05	65.35	4.46	3.1646	3.8742
0.10	48.00	4.37	3.1221	3.8223
0.15	37.82	4.33	3.0876	3.7800
0.20	30.99	4.25	3.0119	3.6873
0.30	22.63	4.10	2.8608	3.5024
0.40	17.50	4.02	2.8105	3.4408
0.50	14.69	3.95	2.7637	3.3834
0.60	12.12	3.87	2.7058	3.3126
0.70	10.36	3.79	2.6470	3.2406
0.80	9.07	3.72	2.5968	3.1792
0.90	7.92	3.65	2.5464	3.1174
1.00	6.98	3.57	2.4898	3.0481

In order to better understand Table 2.8, let us consider an example: when the system has a steady-state velocity of 0.1 $\mathrm{m\,s^{-1}}$ (second line of the table), a variation of 1 bar in the value of p_{B0} (3 bar) results in a velocity variation of:

$$3.8223\text{x}10^{-7} * 1\text{x}10^{5} = 0.0382\ \mathrm{m\,s^{-1}}$$

This corresponds to a relative variation of 38.2%. These relative variations are then represented, for each steady-state velocity, in Fig. 2.13, showing the relative impact that a pressure change has on velocity.

Returning to Table 2.8, we can see that, somewhat unexpectedly, $\left\lvert\frac{\partial \dot{x}}{\partial p_B}\right\rvert_{p_{B0}}$ does not vary that significantly, even though it does decrease with increasing velocity. We can also see how the leak flow ($\dot{m}_{B1}$) compares to the main flow ($\dot{m}_{B2}$). Finally, Fig. 2.13 shows how much impact a pressure change has on the steady-state velocity, depending on its initial value. As expected, that impact decreases significantly for higher velocities.

As a closing note, it should be acknowledged that the problem of coupling is not a critical one in the context of this work - the pressure in the auxiliary chamber is being regulated - but it is undoubtedly worth studying. Being aware of this coupling and how it changes with the system state not only provides insight into the physical

behaviour of the system (which is always good), it can also help with the conceptual part of the controller design, i.e. it can suggest the use of "crossed" control actions in each controller, such as feed-forward velocity terms in the pressure loop. As we'll see in Chapter 3, that is indeed the case.

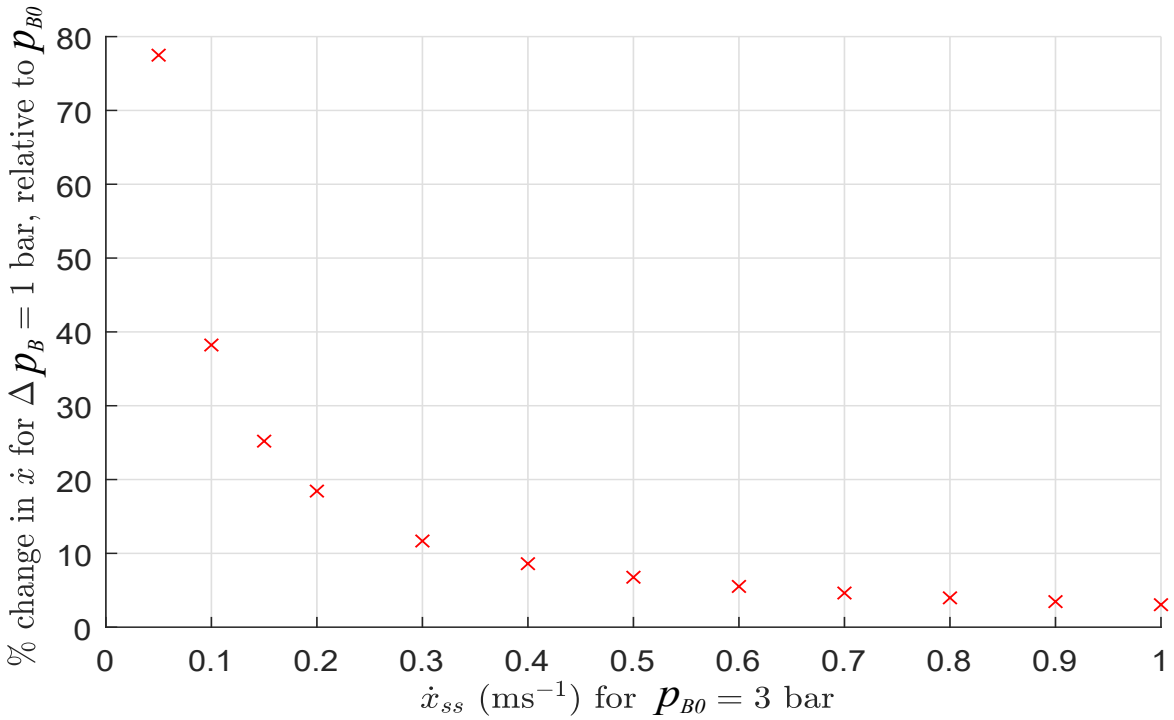

Fig. 2.13 Percentage change in velocity due to changing p_B from 3 to 4 bar, as a function of $\dot{x}_{ss}$.

2.7 Implementation

This section is dedicated to the implementation in MATLAB/*Simulink*® of several elements that aim to make the model a more accurate representation of a real servo-pneumatic system, accounting for the quantization, sampling time of the data acquisition system and electrical noise in the pressure measurements. When added up, these elements often have a non-negligible negative effect on the performance of the controlled system, so they should be taken into account for a proper evaluation of controller performance.

2.7.1 Quantization and Time-discretisation

In the servo-pneumatic system that is modelled in this work, the position values are given by a digital encoder with a finite resolution, and the measurements are updated at a rate defined by the data acquisition system's sampling rate. The control actions and pressure measurements are updated at that same rate. The pressure sensors also have a finite resolution, but in the context of this work, the resulting quantization of those measurements is negligible. Table 2.9 shows the characteristics of the different measurements.

Table 2.9 Characteristics of the different measurements/signals in the servo-pneumatic system

Measurement	Quantization	Sampling Rate	Noise
Position (x)	$5\text{x}10^{-6}$ m	1000 Hz	-
Pressure (p)	-	1000 Hz	Electrical
Control actions (u)	-	1000 Hz	-

2.7.2 Noise Modelling

As was already mentioned, the pressure measurements contain electrical noise, and it's those noisy measurements that are fed back to the controllers, after passing through low-pass filters with a 100 Hz cut-off frequency. Since this noise changes the information that is used by real-life controllers, it should also be properly accounted for in a simulation environment.

The noise can be approximated by gaussian white noise and is characterised by a flat power spectral density (PSD), the height of the PSD being equal to the variance of the process. So we need to know the characteristics of the noise that is observed in the real pressure measurements - for that, we use a large set of actual pressure measurements around ambient pressure, and find the variance of that set of values. We found a variance of:

$$\sigma^2 = 81957.9 \text{ Pa}^2 \tag{2.124}$$

Knowing the variance, using the "*band-limited white noise*" block in *Simulink*® we can generate noise that is a close approximation of the one observed in the real measurements. Because what we have is actually a discrete-time process (samples from a continuous-time process), the value of the variance put into the "*noise power*" option of the aforementioned block must be multiplied by the sampling time, in order to get the correct amplitude (for more information, see documentation of *Simulink*® block "*band-limited white noise*"). The noise is added to the pressure values coming out of the model, and this noisy signal is then passed through a first-order low-pass filter with a cut-off frequency $f_c = 100$ Hz. The filter is modelled by the following transfer function G_f:

$$G_f(s) = \frac{1}{\frac{1}{2\pi f_c}s + 1} \tag{2.125}$$

Equation 2.125 is implemented in block-diagram form, so that the integrator can be properly initialized.

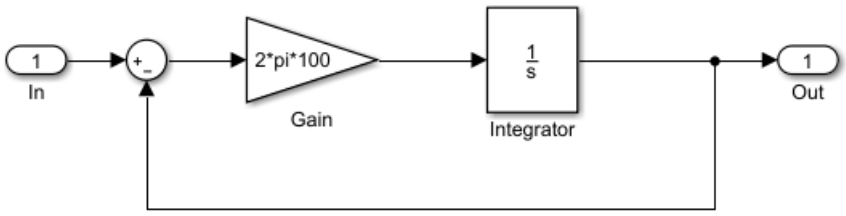

Fig. 2.14 Block-diagram implementation of the first-order low-pass filter

The "*band-limited white noise*" block was then tuned until the high-frequency activity was sufficiently rolled off and the modelled pressure signal closely resembled the actual measurements. Figure 2.15 shows a comparison of modelled and real pressure signals.

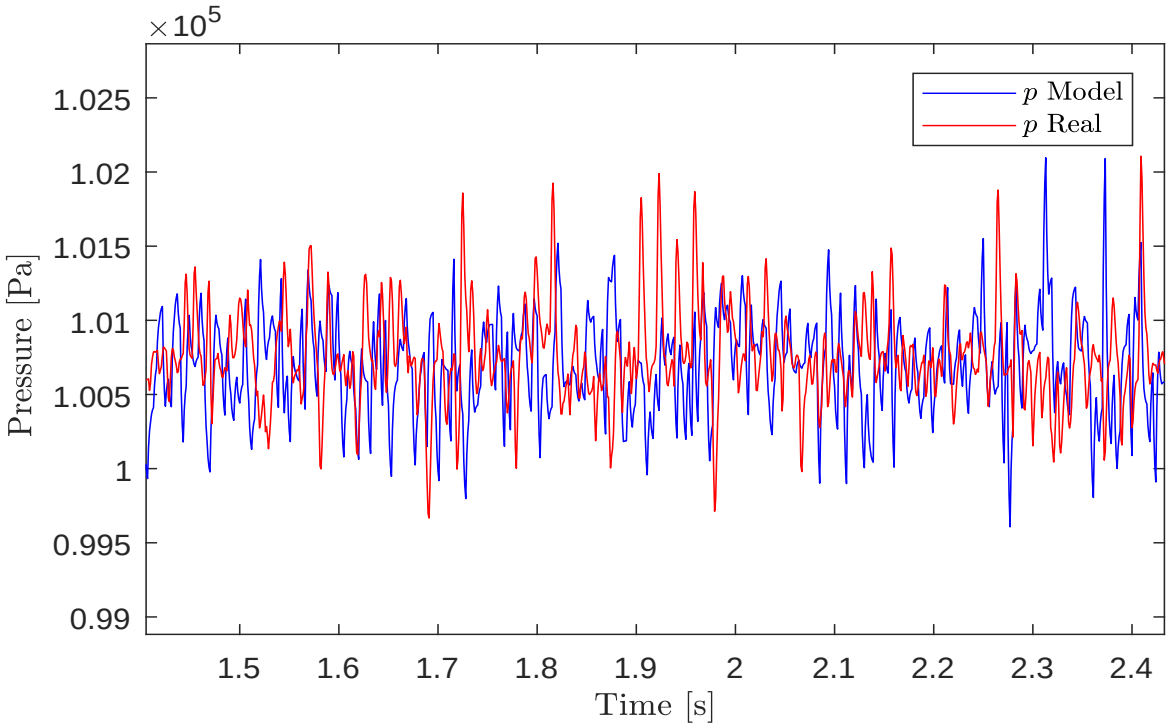

Fig. 2.15 Pressure readings given by the model *vs* the real pressure sensor

2.8 Chapter Conclusions

This chapter opened with the review and update of a 6th-order non-linear model of the servo-pneumatic system under study. The thermodynamic models of the chambers, the servo-valve models, the mechanical model of the actuator and a friction model were all presented, and made up the complete system model.

The servo-valve model was updated to contemplate inverse flow in its restrictions, in the event of the pressure in a chamber rising above p_s or falling below p_{atm}. Concerning friction, a LuGre friction model is presented, since it is deemed to best describe the friction phenomena that are so critical in typical servo-pneumatic applications, most importantly stick-slip motion.

The full 6th-order non-linear model is used for simulation purposes, but it is too complex for analysis or control design purposes. We proceeded to simplify it by assuming the thermodynamic process inside the chambers to be polytropic - we then have algebraic temperature equations instead of differential ones, therefore obtaining a 4th-order system. Linearising it around an equilibrium point, we are presented with a 4th-order linear system model, which allows for a state-space and subsequent transfer function representation. These representations highlighted the coupled nature of the dynamics of pressure and velocity and how the closing of a feedback loop to control one of those variables also influences the other. This coupling of course makes it more difficult to independently control pressure and velocity, and essentially invalidates typical SISO stability analysis tools, such as the root locus, since we have two independent variable inputs.

This coupling was studied from a more practical standpoint, and it was seen how choked-flow conditions can provide a natural decoupling of pressure and velocity, since it implies that volumetric flow-rate, which defines piston velocity, is independent of pressure. Unfortunately, we saw that leak flow through one of the restrictions limited this decoupling effect. It proved to have a significant effect for low velocities, where the leak flow was not at all negligible relative to the "main" flow. However, the relative sensitivity of velocity to pressure variations proved to heavily decrease for higher velocities.

Having gained some insight into the dynamics of the controlled variables, which could then be used in the control design process, we proceeded to discuss the implementation of "real-life" elements in the *Simulink*® model: all values are updated at 1 kHz (frequency of the data acquisition system), position readings are quantized (5 µm encoder resolution) and noise is added to the pressure readings, which are in reality subject to electrical noise.

Chapter 3

Design of a Linear Controller

3.1 Introduction

This chapter will deal with all things related to control design. First, a brief review of classical PID controllers (which will be used as datum controllers) is made. Afterwards, the controller developed throughout this work will be presented. As was already briefly mentioned throughout the text, the controller is based on a two-loop architecture - one for pressure control and another for motion control - utilising the potential of the additional degree of freedom provided by the use of two servo-valves. This architecture will be presented in Section 3.3.

Both the control loops (pressure and motion) rely on linear PID-family controllers to impose the desired dynamics. It is a known fact that, in servo-pneumatics, non-linear controllers typically present much better results than their linear PID counter-parts in a single-degree-of-freedom configuration ($u_A = -u_B$), but they require a more complex design process to avoid the *chattering* phenomenon, and even then still might show too much high-frequency control activity, which is to be avoided if smoothness of movement is an important requirement. Having distinct velocity and pressure control should help with motion smoothness, since not only are we trying to keep the pressure value inside a window that guarantees choked flow through the exhaust port (partially shielding piston velocity from pressure fluctuations), we're also trying to keep it at a constant value of 3 bar.

Speaking in a broader sense, PID controllers in general have the advantage of being a widespread, tried-and-proven solution that has been extensively studied - PID controllers and their variations make up a large percentage of all industrial controllers. They may rarely provide *optimal* performance, but they most often provide *satisfactory* performance for a given application, and prove to be very useful in situations where a plant model is not available, due to excellent understanding of the typical control actions

(proportional, derivative and integral), what their effect on the system is, and empirical tuning methods that allow for satisfactory control even in such a situation where a model is not available. PID control is thus a strongly justified option if heuristic tuning is necessary, allowing us to gradually build the controllers in a simulation environment by using both the knowledge of PIDs and of the system model to introduce the proper feedback loops, account for the more prevalent undesired dynamics and make an educated tuning of the controller parameters.

Concerning expectations, this configuration of separate velocity and pressure loops should ideally provide better accuracy and smoothness than the classical PID architectures, which feature symmetrical control actions. For this reason, a sub-section of this chapter will briefly go over the development of a classical PID controller, to be used as a performance benchmark in Chapter 4.

We then proceed to present the design process for pressure and motion control. This chapter will not go into extensive evaluation of controller performance (that will be done in Chapter 4), though generic results will be shown that allow for a qualitative analysis of issues such as the presence of limit-cycles and the general stability of the system. A section will be dedicated to the issue of controller switching, since the servo-valve controllers should switch roles when the direction of motion is changed.

Unless stated otherwise, we will always be referring to forward movement, where the main chamber is chamber A and the auxiliary chamber is chamber B. This can be done without loss of generality. All results presented in this chapter are for a payload mass $m = 4$ kg, and S-curve (sigmoid) inputs. It is much smoother than a step-input and helps in reducing overshoot, so all controllers were tuned for optimal response to S-curve inputs.

3.2 PID Control

3.2.1 Basic Control Actions

Before detailing the control design process for the motion and pressure loops, a brief review of PID control will be made.

The PID controller uses, of course, the simple yet powerful concept of feedback: a certain process variable is monitored and fed back to the controller, which then compares the value of the process variable with a reference value. The control algorithm then produces a control action that, through an actuator, tries to impose the necessary dynamics on the system (within its real physical capabilities) that result in the process variable getting closer to the reference value.

The simplest form of a feedback controller is the proportional controller, which is very basic in principle: the further the process variable is from the desired value (i.e. the larger the error), the stronger the control action is - hence the term "proportional". But it is easy to see that a simple proportional controller has many shortcomings. It does not account for either the trend of the error or its history. In other words, there is neither an attempt to predict the future of the process variable, nor is there a "memory" of its past.

The PID controller does all of those things. The resulting control action has a term proportional to the error, a term proportional to the derivative of the error (accounting for its rate of change), and a term proportional to the integral of the error, this one accounting for its history. The "textbook" PID control action is thus given by:

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{d}{dt}e(t)\right) \tag{3.1}$$

where K is a gain, e is the error between the desired value and the actual value, and T_i and T_d are the so-called integral time and derivative time, respectively. In the frequency domain, equation 3.1 turns into:

$$U(s) = KE(s)\left(1 + \frac{1}{T_i s} + T_d s\right) \tag{3.2}$$

The various terms of the PID control action have different effects on a system's dynamics. Although those effects depend on the system itself and on the tuning of the controller parameters, a general outline can be provided:

- The **proportional** action has the typical effect of improving response-time and lowering steady-state error, but at the cost of an increasingly oscillatory response as the proportional gain is raised.

- The **integral** action is used to reject low-frequency disturbances , and is mainly responsible for improving/eliminating the steady-state error. It can be seen as a way to account for the "permanent disturbance" that is steady-state error. For $T_i = \infty$, the integral action becomes null. For large values of T_i, the response slowly creeps towards the reference, while for smaller values the response is faster but also more oscillatory.

- The **derivative** action is used to introduce damping into the system and improve closed-loop stability by estimating future error. $T_d = 0$ makes for null derivative action. Damping of the response initially increases with T_d, but if its value is set too large, the response may become oscillatory.

The aforementioned control actions may also be directly applied to the output variable rather than to the error. The nomenclature used in this work will distinguish those cases by using a dash (**-**) before the corresponding control action. This is similar to the nomenclature present in Ogata (2001).

The following sub-section will present a classical architecture PI-D controller, to be used as a performance benchmark. In that controller, proportional and integral actions act on the error whilst the derivative action acts directly on the output variable. Sections 3.3.2 and 3.3.3 will then provide insight into the design process of PID-family controllers for pressure and motion control. It is important to note that the controllers (pressure and motion) were developed mainly in parallel, but for the sake of structure we will first discuss pressure control, followed by motion control. This means that even though pressure control comes first, some allusions to motion control have to be made to justify development decisions.

3.2.2 Datum Controller

In order to get the fairest comparison, three different position controllers were briefly tested with the classical $u_A = -u_B$ architecture: a PID with filtered derivative, a PI-D and an I-PD, all of them with anti-windup mechanisms and a dead-zone in the integrator. Of these three, the one that presented better performance was the PI-D. The I-PD presented almost no overshoot (as expected), but it lagged behind and the piston moved after being stationary for some time. The PID had a more similar performance to the PI-D, but the latter usually presented smaller overshoot values.

We will therefore use the PI-D as a benchmark for the tests. Fig. 3.1 illustrates a PI-D with a classical architecture, with a dead-zone and anti-windup.

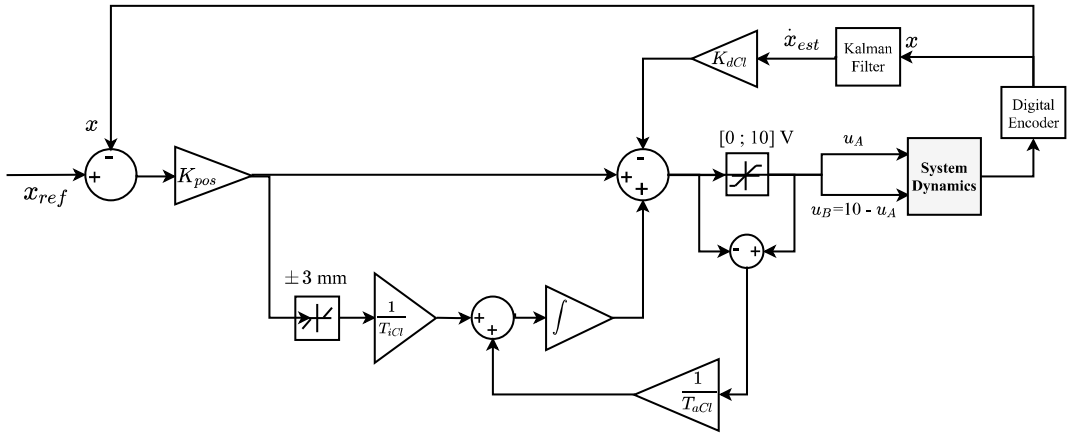

Fig. 3.1 Classical architecture PI-D controller

Table 3.1 contains the controller parameters.

Table 3.1 Classical architecture PI-D parameter values

Contr. Parameters	K_{pid}	K_{dCL}	T_{iCL}	T_{aCl}	Dead-Zone
Values	85	3.9	0.12	0.04	±3 mm

3.3 Development of the new controller

3.3.1 Proposed Architecture

In the proposed architecture, the pressure in the auxiliary chamber is fed back to the pressure controller, which is actually connected to the main chamber's servo-valve. This servo-valve acts so as to maintain p_{aux} at a constant value of 3 bar, thus acting like a regulator. The other servo-valve, connected to the auxiliary chamber, acts according to the motion specifications of velocity and position. In steady-state conditions, it essentially tries to emulate a sort of meter-out configuration, limiting the flow rate of air exiting the chamber and thus controlling piston velocity. To accelerate the piston, the servo-valve connected to the auxiliary chamber opens up to the atmosphere, lowering the pressure in that chamber and resulting in a larger pneumatic force differential between chambers. For deceleration, that same servo-valve allows for pressurised air to enter the chamber, acting as a brake to the piston's momentum. This velocity controller will be part of a cascaded motion controller - the input of the velocity controller is the output of a proportional positioning controller. Fig. 3.2 shows a simplified representation of the control architecture. The switching of the controllers for backward (BWD) and forward (FWD) motion is discussed in Section 3.3.4, and the full detailed version of the controllers will be presented in section 3.3.5.

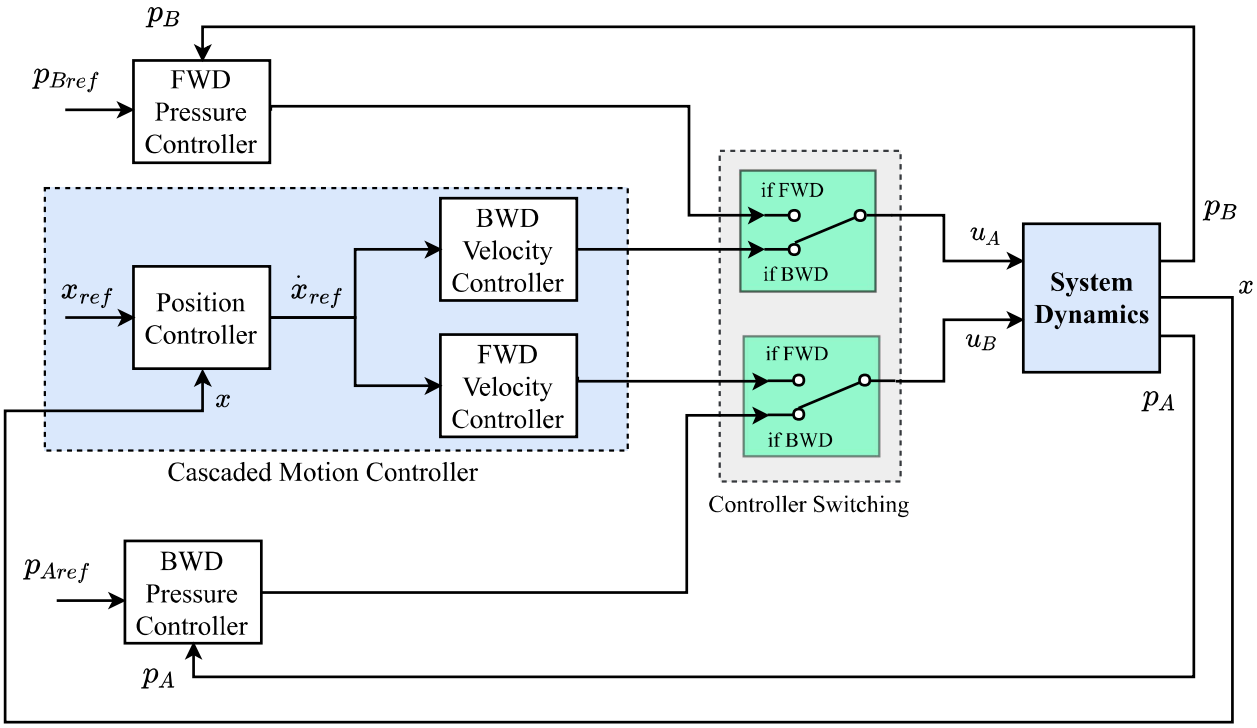

Fig. 3.2 Simplified schematic representation of the control architecture

3.3.2 Pressure Control

The pressure control-loop aims to indirectly control the pressure in the auxiliary chamber (p_B, unless stated otherwise) by using servo-valve A to act on p_A. The controller essentially acts as a regulator, trying to maintain pressure at a constant value of $p_B = 3$ bar, while of course still allowing the pressure variations needed for the system to present satisfactory dynamic behaviour.

Therefore, the requirements on pressure control are not very strict - the ultimate goal of the overall control system is to provide accurate, smooth motion control. The pressure controller should therefore always help - not hinder - motion control, and this of course means that the motion controller should be able to change p_B, for the sake of a good dynamic response to motion specifications, without too much "protest" from the pressure controller. In other words, since it is connected to chamber B, the velocity controller will obviously have a strong impact (as it should) on the pressure inside that chamber, because - after all - a pressure differential is the driving force through which the desired motion is achieved.

Controller Design

The first thing to discuss should be the already mentioned reference of $p_B = 3$ bar. This value is chosen because it poses as a good middle-ground between p_{atm} and p_s, as the pressure ratios will generally lie under the critical pressure ratio (around 0.5 (Falcão Carneiro, 2007)), promoting choked flow and its underlying advantages for motion control. Furthermore, these lower than usual steady-state pressures also help in improving general smoothness. It is true that this is due to a decrease in the pneumatic stiffness of the system, which in turn affects its bandwidth, but a slightly lower responsiveness to abrupt changes can be seen as a good thing in terms of smoothness.

Let us now proceed with the discussion of the actual controller. As was already said, there are no strict specifications that the controller must meet - it should be able to bring p_B relatively quickly to a steady-state 3 bar value and, once the desired position is achieved (with a given error), it should be able to compensate for natural pressure dynamics that would otherwise displace the piston after it has already stopped. This last point is very important, because as we'll see, that is indeed a problem.

We begin by establishing a primary building block: a **PI** controller, the most used PID-family variation. It solves the P-only controller's main problem, which is steady-state error, while avoiding potential problems that the derivative action may cause due to differentiation of a noisy signal, which is indeed the case. Besides, the damping effect of derivative action is not quite necessary in this loop.

Another *a priori* design decision that was made relates to what we have learned about the system in Chapter 2, specifically in the discussion about the velocity and pressure coupling. We know that if, for example, the motion controller requests a decrease in velocity, then servo-valve B is going to act so as to increase p_B to brake piston momentum. So what we can do is use this information and feed it "directly" to the pressure controller, bypassing the system's dynamic response. What this means is that we'll be using a feed-forward velocity term, using the velocity reference. This way there is a "prediction" term in the control action of the main chamber. An increase in p_B is expected from a decrease in $\dot{x}_{ref}$ - the velocity feed-forward term will then decrease immediately with $\dot{x}_{ref}$, reducing the overall u_A control action and helping in reducing piston velocity.

Controller: PI + Velocity feed-forward

So up until now we have a PI controller with a feed-forward velocity term. To get some insight into how the controller performs in the context of position control, we take a jump forward and present some positioning results, using the final version of the velocity controller (to be discussed later) and the current version of the pressure controller. As we can see in Fig. 3.3, a limit-cycle occurs.

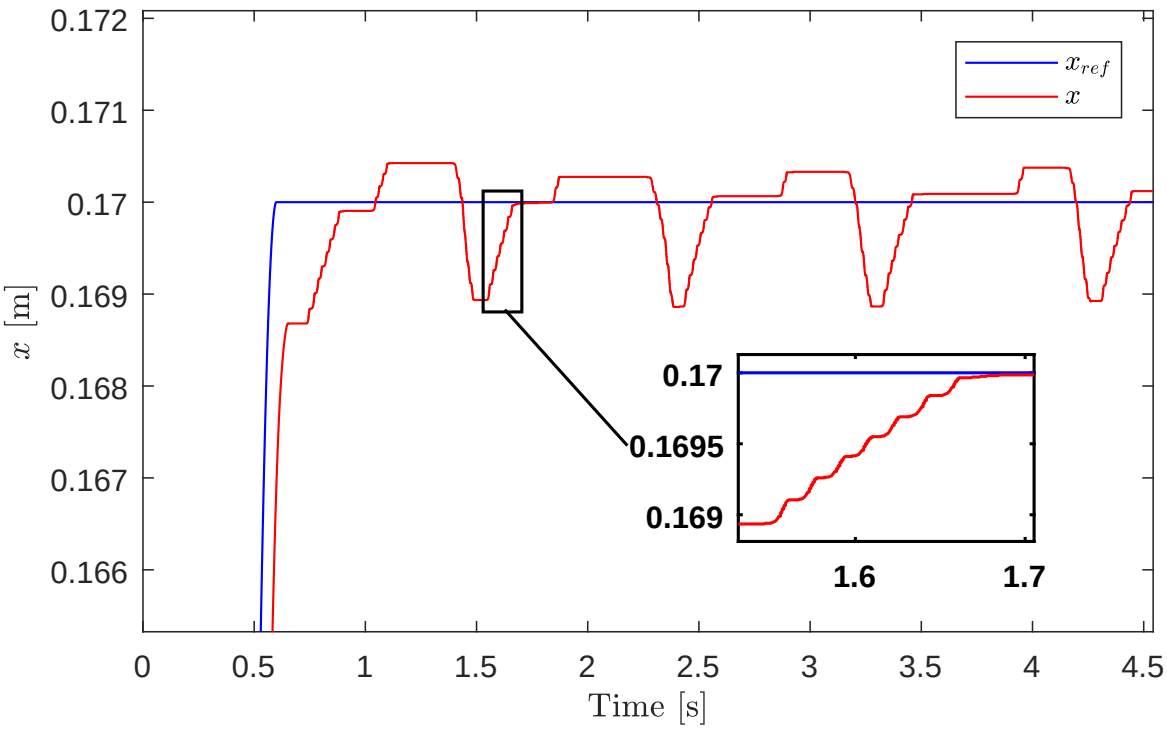

Fig. 3.3 Positioning test using a PI pressure controller with feed-forward velocity term.

This is called a hunting limit-cycle - the system "hunts" for the desired position but is never quite able to reach it. This happens due to the combination of friction and integral action, resulting in the system ending up in a stick-slip oscillation around the desired position (Hensen et al., 2003). Since the velocity controller has no integral action (see section 3.3.3), we know this is due to integral action in the pressure loop.

The pressure in the auxiliary chamber is shown in Fig. 3.4. We can see that after the initial dynamic response, the controller is not able to achieve a steady-state value. The high-frequency pressure variations translate into the stick-slip small displacements shown in the zoom-box of Fig. 3.3.

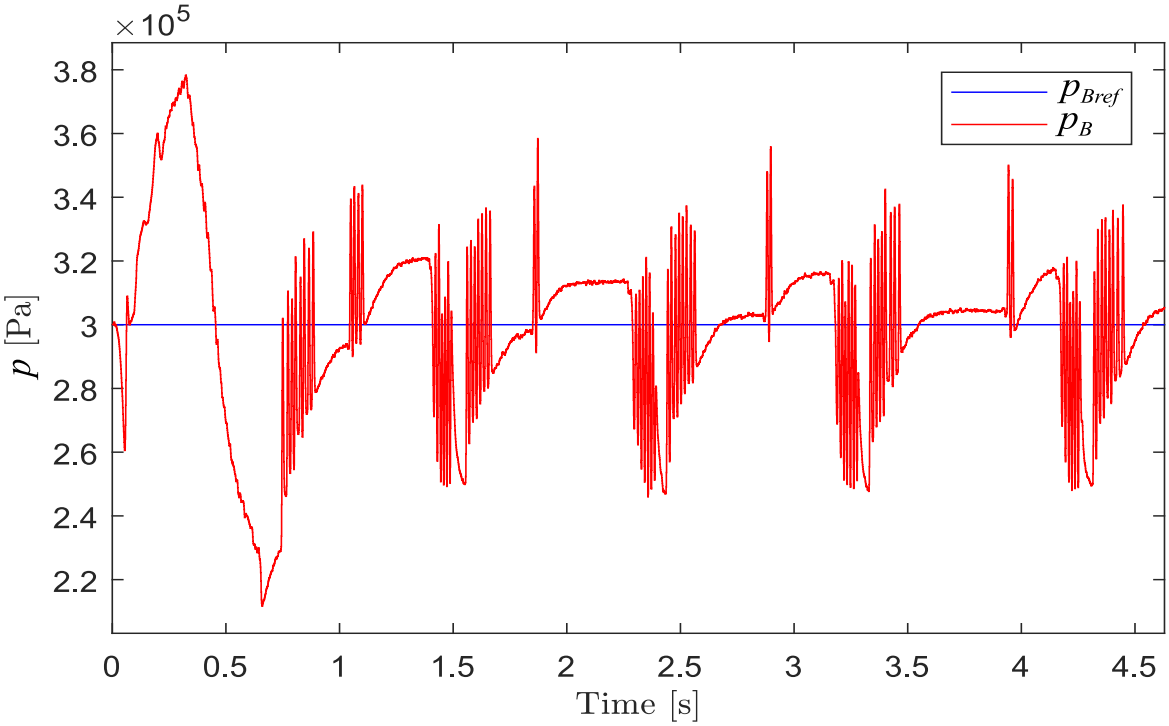

Fig. 3.4 Evolution of p_B for the test shown in Fig. 3.3

Controller: PI + Velocity feed-forward + Integrator-Freeze

Typically, the limit-cycle can be eliminated by introducing a so-called dead-zone - an error window where the integral action is turned off, therefore avoiding the appearance of the hunting phenomenon at the cost of a higher steady-state error. However, what we'll do in this situation is slightly different, because we will monitor the position error instead of the pressure error, so that the pressure integrator freezes if the absolute value of the position error goes below a certain value. This means that while the position error lies below that value, the integral control action is kept constant.

Implementing this integrator freeze, we can eliminate the hunting phenomenon, as shown in Fig. 3.5.

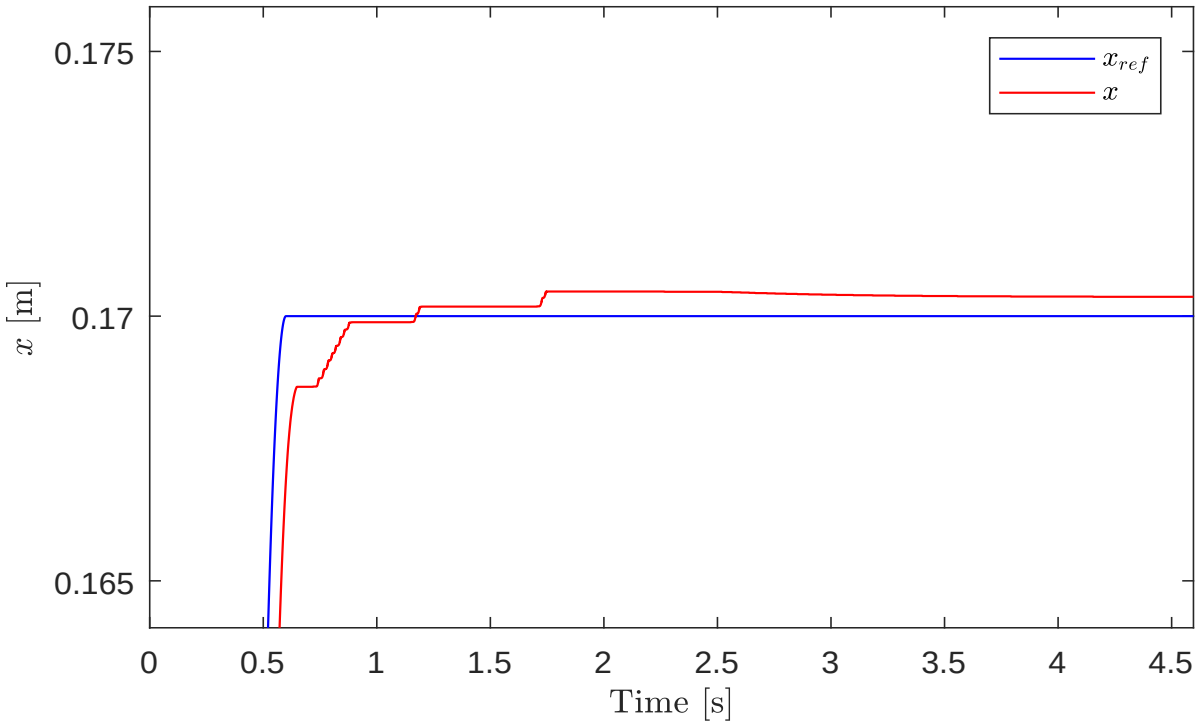

Fig. 3.5 Positioning test with integrator freeze.

A side-effect of the integrator freeze is that the steady-state pressure value will be dictated by the value of p_B when the freeze-zone is reached, and so in many situations it will stabilize around values other than 3 bar, although not very far. For the same positioning test of Fig. 3.5, we can see the evolution of p_B in Fig. 3.6.

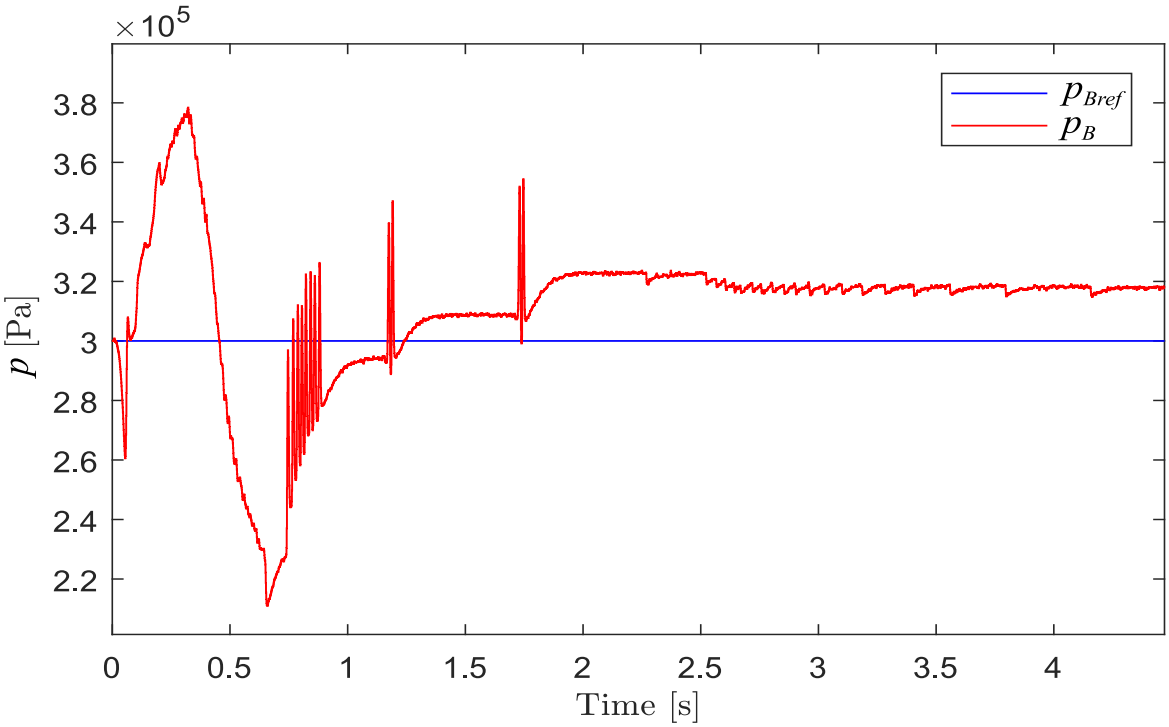

Fig. 3.6 Evolution of pressure for positioning test of Fig. 3.5

We have thus found a way to avoid hunting limit-cycles, but note that p_B has a steady-state value of around 3.2 bar, rather than 3 - this is a small price to pay for the avoidance of limit-cycles. There was, however, still another problem: after some positioning movements, a phenomenon was observed in which the piston would slightly move in the same direction of the previous movement, after having been stationary for a small amount of time. The error would then become larger than the limits of the freeze-zone and the piston would jerk back. This phenomenon did not present an inherently cyclic nature. Several examples are shown in Fig. 3.7.

Upon further investigation, it was seen that this was due to p_A rising after the piston stopped. The pressure in chamber A creeps up until it is high enough to push the piston, and this happens because in this situation p_A has not yet reached the equilibrium value that corresponds to the steady-state u_A defined by the controller. The control action is constant - the integrator is frozen, the reference velocity is zero and the positioning error is not changing - but p_A is still moving towards its equilibrium value (different than p_{A0}). Sometimes, the "new" equilibrium p_A value will be large enough to overcome static friction and move the piston. Fig. 3.8 shows the evolution of p_A in that situation.

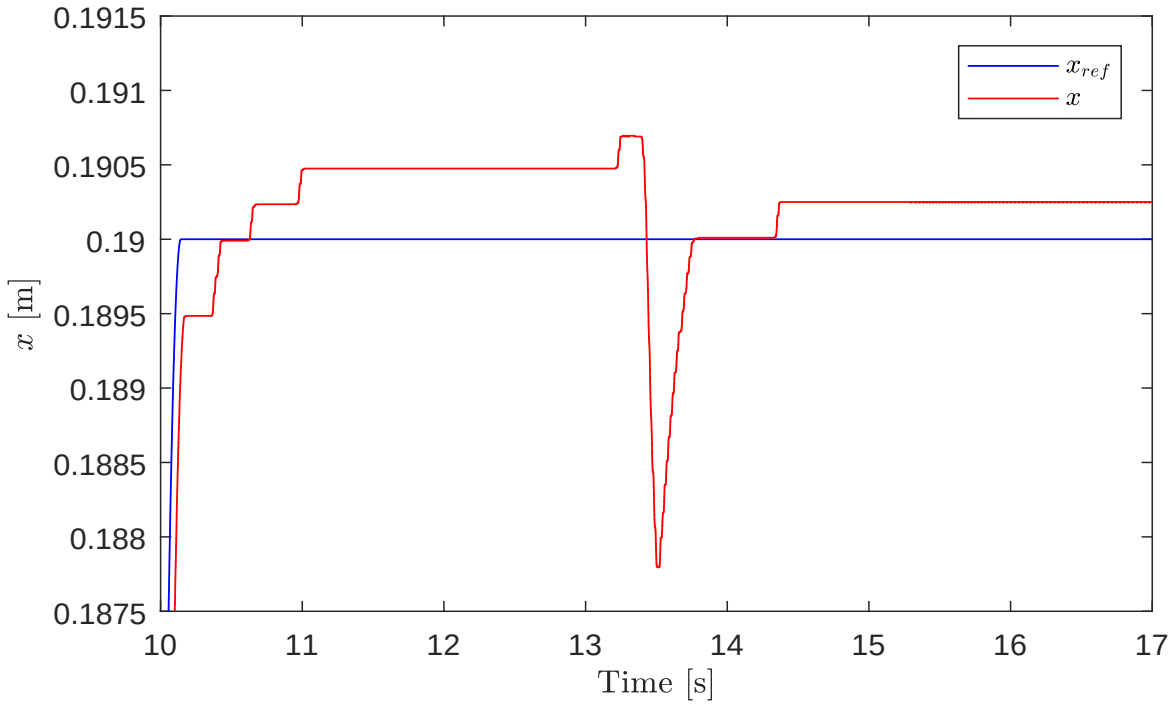

Fig. 3.7 Positioning problem with PI + velocity feed-forward + integrator freeze.

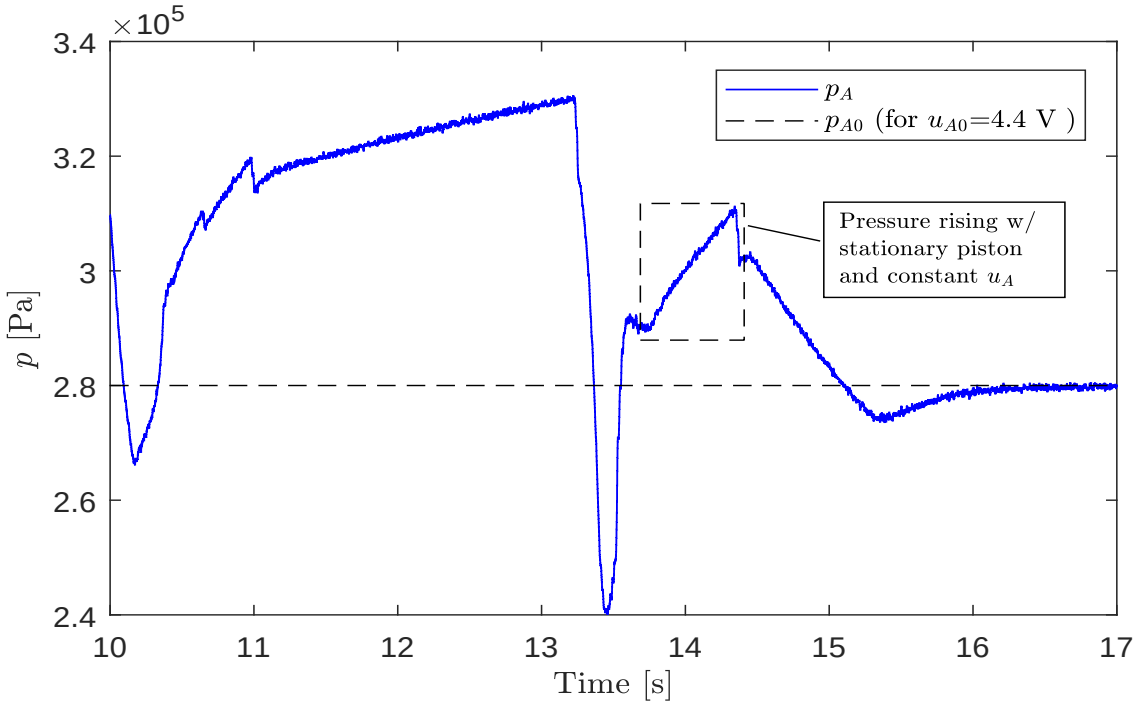

Fig. 3.8 Evolution of p_A for the positioning test shown in Fig. 3.7, with a highlighted example of the problem with unobserved pressure dynamics.

We can see in Fig. 3.9 how the steady-state value of u_A is higher than u_{A0} (for the standard p_{A0} of 2.8 bar), in the period of [13.8 14.3] s. Although the control action remains constant, Fig. 3.8 shows p_A still rising towards its equilibrium value, and the controller does not account for this phenomenon.

This is an observability problem - the controller does not observe the pressure in the main chamber (p_A for forward movement), and is therefore unable to compensate in any way for undesired dynamics relating to that variable. Only when these dynamics translate into an observable change (such as the piston moving) does the controller react. This observability issue is a known problem in servo-pneumatics (Falcão Carneiro and Gomes de Almeida, 2011), and was named *sticking and restarting phenomenon* (SRP) in the work by Brun et al. (2005), where it was more extensively studied.

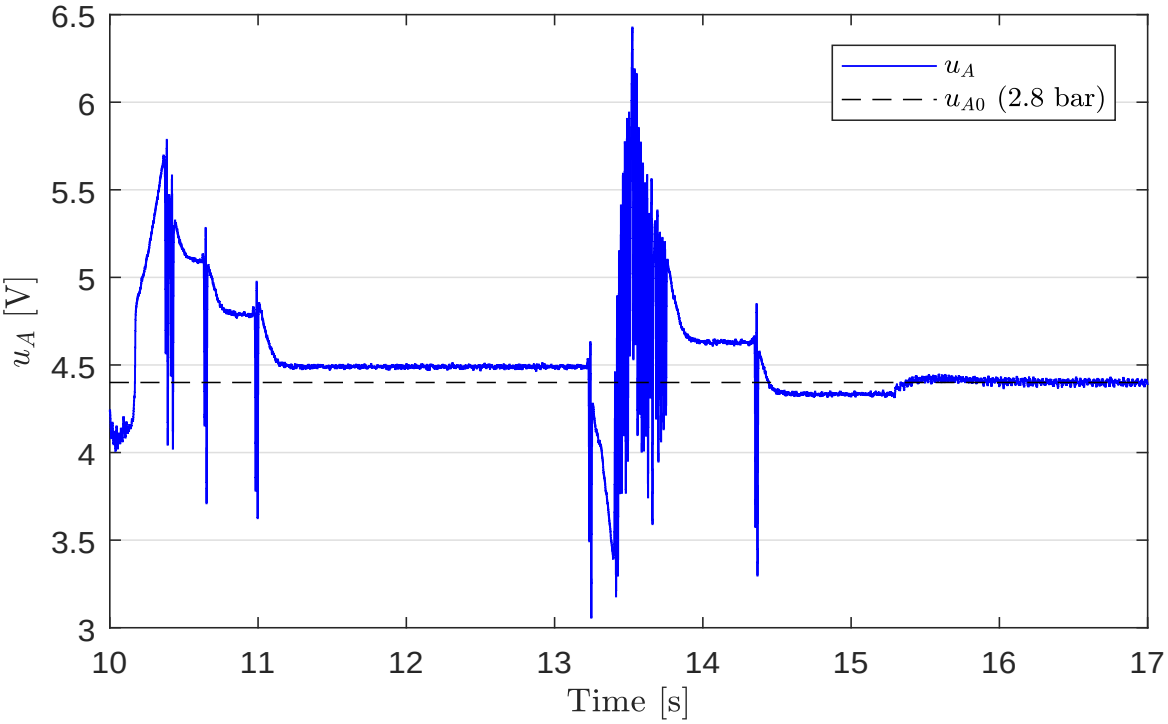

Fig. 3.9 Control action u_A for the positioning test shown in Fig. 3.7

Controller: PI + Velocity feed-forward + Integrator-Freeze + p_A term

Up until now, the controller did not account in any way for the pressure dynamics in chamber A, although the value of p_A is obviously accessible through the pressure sensor of that chamber. We can therefore find a way for the controller to monitor p_A and try to compensate for its rise, solving the observability problem and leading u_A towards a value that is closer to u_{A0}. This can be done by introducing a control action term that is proportional to p_A, and subtracting it from the total control action. After

implementation and tuning of the p_A term, the problem is successfully eliminated, as shown in Fig. 3.10. Fig. 3.11 shows that we now have $u_A \approx u_{A0}$ in steady-state.

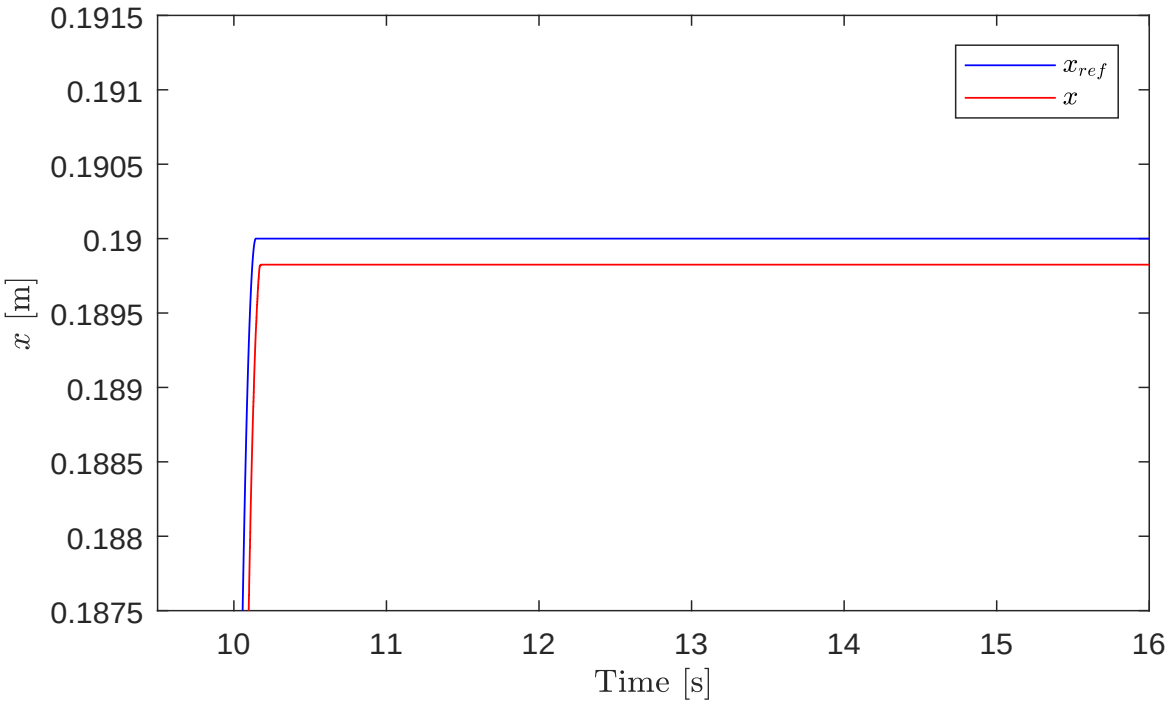

Fig. 3.10 Positioning test with PI + $\dot{x}_{ref}$ feed-forward + integrator freeze + p_A term.

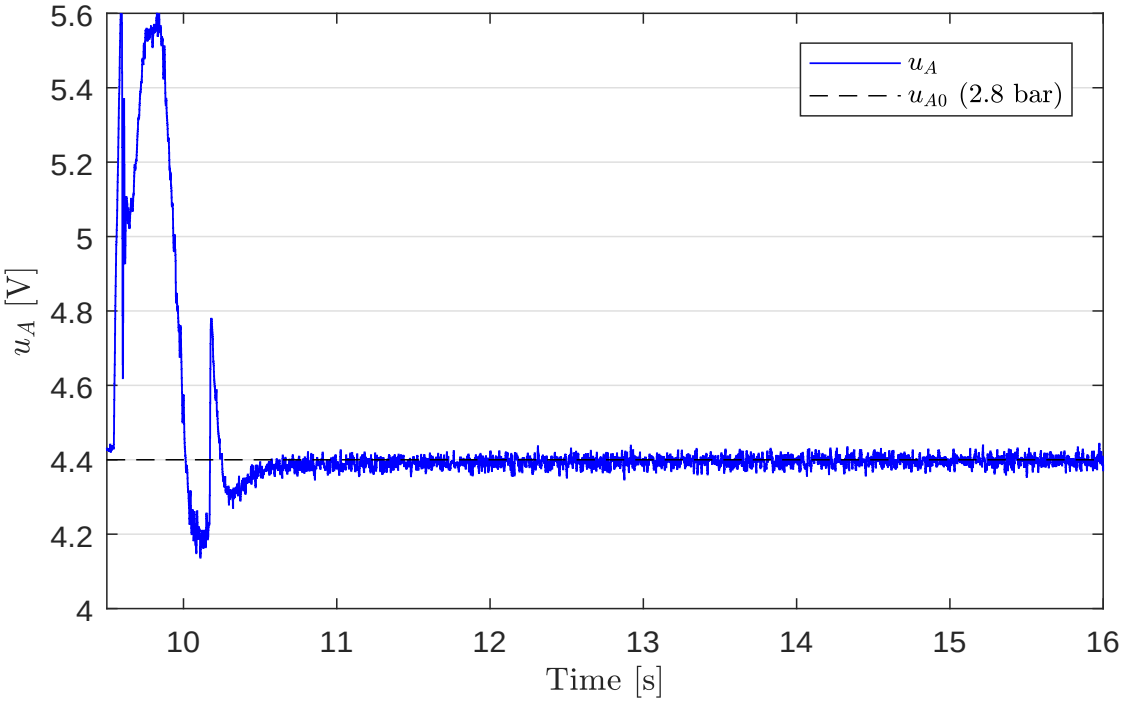

Fig. 3.11 Control action u_A for the positioning test shown in Fig. 3.10

Anti-Windup

Windup is a phenomenon caused by the interaction of integral action and saturations. All actuators have physical limits that they cannot surpass: in our case, restriction R_1 of the servo-valves, for example, cannot be more than fully opened (10 V) or fully closed (0 V), so it may happen that in some situations (usually a large change of a reference value), the control action reaches the actuator limits - it becomes saturated. When that happens, if there is integral action, the error continues to be integrated and the overall control action keeps getting farther away from the saturation limit. This means that when the error starts decreasing, the integral value will still be very large, preventing the controller from quickly resuming "normal" operations - it takes time for the integral value to decrease to within the saturation limits (Åström and Hägglund, 1995).

An anti-windup mechanism can be implemented to deal with this problem. That mechanism monitors the difference between the saturated value and the actual value of the control action ($u_{sat} - u$), and then subtracts the difference (multiplied by a tunable constant) from the integral value, therefore keeping it close to the saturation limit. This means that when the error decreases, the integral value can quickly go back inside the saturation limits and into normal working conditions. The implementation of this anti-windup mechanism can be seen in Fig. 3.12. We can see that the tunable constant (we will refer to it as T_a) can be interpreted as a time constant, defining how quickly the integrator resets after the saturation limits are surpassed.

It should be noted, as an *a posteriori* observation, that in the case of the controllers used in this work, anti-windup is only necessary for step inputs. Using S-curve inputs, large abrupt changes in the reference values are avoided, and thus control actions exceeding the saturation limits were never observed.

Final Pressure Controller

The functional blocks of the final pressure controller are represented in Fig. 3.12, now only for forward motion. Section 3.6 will present the final motion and pressure controllers for both forward and backward motion.

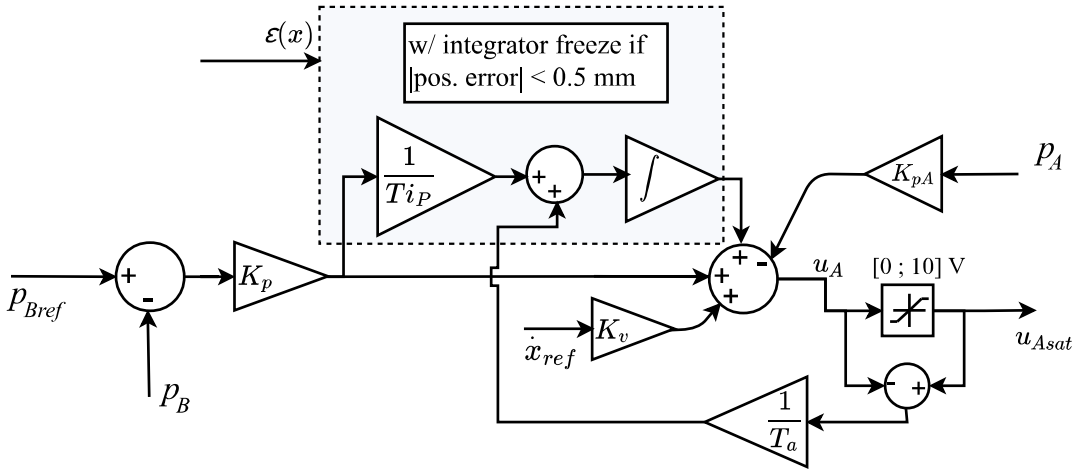

Fig. 3.12 Schematic representation of the final pressure controller for forward motion.

For the sake of clarity concerning the integrator freeze, we will now present the control action equations (Eqs. 3.3 and 3.4) for the different working conditions. Let ε_x be the positioning error, and u_{frz} the integral control action value at the instant in which the system enters the integrator freeze-zone. u_{int} is the integral control action, u_{pA} is the control action proportional to the pressure in the main chamber, u_{prop} is the usual proportional control action and u_{FF} is the velocity reference feed-forward term. The pressure-loop control action will then be given, for forward motion, by:

If $|\varepsilon_x| \geq 0.5$ mm :

$$u_A(t) = u_{int}(t) + u_{pA}(t) + u_{prop}(t) + u_{FF}(t) \tag{3.3}$$

If $|\varepsilon_x| < 0.5$ mm :

$$u_A(t) = u_{frz} + u_{pA}(t) + u_{prop}(t) + u_{FF}(t) \tag{3.4}$$

The values of the controller parameters are shown in Table 3.2, considering pressure values given in Pa. The parameter values are the same for the backwards-movement version of the controller.

Table 3.2 Pressure controller parameter values

Contr. Parameters	K_P	Ti_P	K_v	K_{pa}	T_a	Freeze-Zone
Values	2×10^{-5}	0.09	10	2×10^{-5}	0.35	$\pm$0.5 mm

3.3.3 Motion Control

As was already mentioned in the previous section, the motion and pressure controllers were essentially developed in parallel. When a problem arose with the performance of one of the controllers, sometimes the problem (or rather the solution to that problem) did not lie in that controller. In other words, certain developments on a given controller sometimes created the need for small tweaks, or even fundamental changes to the other one, in order to improve overall performance. It is this back-and-forth nature of the design process that makes it hard to present it in a linear fashion, and that is why some references are made to the motion controller in section 3.3.2 - the performance of the individual controllers should always be discussed in the context of the overall objectives of this work, not in a vacuum.

We have thus already discussed the origin and the solution of some major motion control problems that originated in the pressure control loop, namely:

- The presence of a hunting limit-cycle, due to the non-linear nature of friction associated with the integral control action. This was solved by implementing an integrator freeze as a function of positioning error: while the positioning error remains smaller than a pre-defined value, the output of the integrator will be held constant and equal to its value at the instant it entered the freeze-zone.

- The pressure creep in the main chamber while the piston was stationary, which resulted in undesired small movements. This proved to be an observability problem: since u_A could settle with a slight offset relative to u_{A0}, the equilibrium value of p_A would also differ from p_{A0}, and there would be a period of time where the piston was stopped and u_A was constant, while p_A was still moving towards its new equilibrium value, without the controller being able to compensate for it. On the way to this new equilibrium condition, the pressure would often reach a value that was enough to overcome static friction and cause undesired movements. This observability problem was solved by introducing a control action term that was proportional to p_A.

This section will further explain the motion control design process, but still in a predominantly qualitative manner. Before that, since the controller needs to have access to velocity and acceleration values but we can only measure position, an overview is given of a state estimator - the Kalman filter - which will provide access to accurate estimations of velocity and acceleration, enabling the use of feedback loops involving those variables.

The Kalman Filter as a Velocity and Acceleration Estimator

The Kalman filter is a linear quadratic estimation (LQE) algorithm that combines sensor measurements (assumed to contain noise and other inaccuracies) and a system's dynamic model to produce estimates of the system states. In the context of this work, the Kalman filter can be used to obtain estimates of velocity and acceleration, whose values would otherwise be inaccessible (or only accessible through inaccurate methods).

The Kalman filter used in this work had already been previously developed, we needed only to implement it in the controller. A brief overview of the discrete Kalman filter algorithm will therefore be provided.

The discrete Kalman filter follows a simple concept: it produces an estimate of the system state based on the system model, and then corrects that estimate based on new measurements, i.e. the algorithm features two steps - estimation and correction. It is therefore divided into estimation equations and correction equations. The discrete estimation equations are given by:

$$\hat{\mathbf{x}}_t = \mathbf{A}\bar{\mathbf{x}}_{t-1} + \mathbf{B}\mathbf{u}_{t-1} \tag{3.5}$$

$$\hat{\mathbf{P}}_t = \mathbf{A}\bar{\mathbf{P}}_{t-1}\mathbf{A}^T + \mathbf{Q} \tag{3.6}$$

where:

- $\mathbf{A}$ is the state-transition matrix, in this case given by the state-space representation of the mechanical model.
- $\mathbf{B}$ is the input matrix.
- $\bar{\mathbf{x}}_{t-1}$ is the corrected state vector of the previous time sample.
- $\mathbf{u}_{t-1}$ is the system input for the previous time sample.
- $\hat{\mathbf{x}}_t$ is the estimated state vector for the current time sample.
- $\bar{\mathbf{P}}_{t-1}$ is the corrected error covariance matrix for the previous time sample.
- $\hat{\mathbf{P}}_t$ is the estimated error covariance matrix for the current time sample.
- $\mathbf{Q}$ is the process noise covariance matrix.

The correction equations are given by:

$$\mathbf{K}_t = \hat{\mathbf{P}}_t \mathbf{C}^T (\mathbf{C}\hat{\mathbf{P}}_t \mathbf{C}^T + \mathbf{R})^{-1} \tag{3.7}$$

$$\bar{\mathbf{x}}_t = \hat{\mathbf{x}}_t + \mathbf{K}_t(\mathbf{y}_t - \mathbf{C}\hat{\mathbf{x}}_t) \tag{3.8}$$

$$\bar{\mathbf{P}}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{C})\hat{\mathbf{P}}_t(\mathbf{I} - \mathbf{K}_t\mathbf{C})^T + \mathbf{K}_t\mathbf{R}\mathbf{K}_t^T \tag{3.9}$$

where:

- $\mathbf{C}$ is the measurement matrix.
- $\mathbf{K}_t$ is the Kalman gain matrix for the current time sample.
- $\mathbf{y}_t$ is the measurement vector of the current time sample.
- $\bar{\mathbf{x}}_t$ is the corrected state vector for the current time sample.
- $\bar{\mathbf{P}}_t$ is the corrected error covariance matrix for the current time sample.
- $\mathbf{I}$ is the identity matrix.
- $\mathbf{R}$ is the measurement noise covariance matrix.

The state vector $\mathbf{x}$, state-transition matrix $\mathbf{A}$ and input matrix $\mathbf{B}$ are given by:

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \tag{3.10}$$

$$\mathbf{A} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.12}$$

where T_s is the sample time, $\mathbf{A}$ is derived from the mechanical model of the system and $\mathbf{B}$ is null because there are no known inputs to the mechanical system model. Since we can only measure the position of the piston, the measurement matrix is given by:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{3.13}$$

The noise matrices $\mathbf{Q}$ and $\mathbf{R}$ provide the algorithm with information about the statistical nature of the process and measurement noise, respectively. That information is used to compute the error covariance matrix $\mathbf{P}$. After the estimated values are obtained, the corrected state-vector is computed as a weighted average of the system model estimation and the new measurement. The $\mathbf{K}_t$ gain provides more or less weight to the measurement, depending on the estimated uncertainty of the predicted values, so that those with a lesser uncertainty are more trusted. The covariance represents a measure of that uncertainty in the state predictions. For information on how the $\mathbf{Q}$ and $\mathbf{R}$ matrices were obtained for this servo-pneumatic system, see Carneiro and Almeida (2016).

An important feature of the Kalman filter is that, in the case of time-invariant $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{Q}$ and $\mathbf{R}$ (which is the case), the Kalman gain matrix $\mathbf{K}_t$ quickly converges to a constant value $\mathbf{K}_{ss}$ (Simon, 2006). The Kalman gain can thus be calculated *off-line*, reducing the computational requirements of the algorithm. The steady-state values are given by equations 3.14 and 3.15, the first being an algebraic Riccati equation.

$$\mathbf{P}_{ss} = \mathbf{A}\mathbf{P}_{ss}\mathbf{A}^T - \mathbf{A}\mathbf{P}_{ss}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{ss}\mathbf{C}^T + \mathbf{R})^{-1}\mathbf{C}\mathbf{P}_{ss}\mathbf{A}^T + \mathbf{Q} \tag{3.14}$$

$$\mathbf{K}_{ss} = \mathbf{P}_{ss}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{ss}\mathbf{C}^T + \mathbf{R})^{-1} \tag{3.15}$$

After implementation, the Kalman filter is able to provide accurate estimations of velocity and acceleration that can be used by the controllers. Fig. 3.13 illustrates the Kalman filter's function in the context of this work.

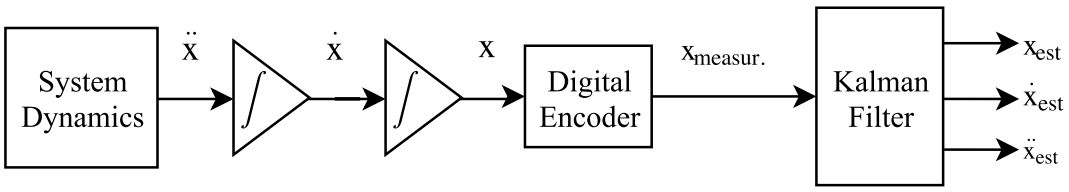

Fig. 3.13 Schematic representation of the Kalman filter implementation.

Controller Design

The previous sub-section explained how the Kalman filter can provide the controller with accurate estimates of velocity and acceleration, something which is of course crucial to the implementation of the necessary feedback loops. Having resolved that issue, we can now proceed with the discussion of the control design *per se*. It should be noted that the tuning of the controllers henceforth presented is done heuristically, in a simulation environment.

Recall that we have already discussed some motion control problems in section 3.3.2, having reviewed them at the beginning of this section. Those problems, and their solutions, were presented resorting to the final version of the motion controller, but as might be expected, this controller also underwent a somewhat iterative design process.

The first iteration of the velocity controller was an **I-PD**, a modification of the PID controller where the error is only used for the integral action - the P and D terms were proportional to the estimated values $\dot{x}_{est}$ and $\ddot{x}_{est}$, respectively. This iteration seemed satisfactory for velocity control at a given point in the design process, but when the position loop was closed and positioning performance was evaluated, it proved to be too problematic. Furthermore, in conjunction with the final version of the pressure controller (which was different when the I-PD was being tested), the I-PD did not present satisfactory performance. The results of this iteration are not deemed relevant to the description of the design process, but it is worth discussing since it presents further proof of the difficulties that arise due to the combination of a PID's integral action and the system's non-linear friction phenomena.

In an attempt to avoid the appearance of hunting limit-cycles, different combinations of dead-zone and integrator-freeze were tried, but in the end it proved too difficult to guarantee stability and satisfactory performance across the whole cylinder's stroke length.

Accuracy in the steady-state velocity value is not critical in the context of position control - it is not a problem if the steady-state velocity slightly differs from the reference value, if this does not translate into a loss of positioning performance. This means that the integral action can, in principle, be removed from the velocity controller, eliminating the aforementioned problems with limit-cycles. That was exactly what was tried, and therefore a **PD** controller was implemented as a replacement to the I-PD. The PD controller, as its name suggests, consists of only proportional and derivative control actions.

Velocity Controller: P-only; Position Controller: P-only

The use of derivative control action should be properly justified, since differentiation of noisy feedback signals tends to amplify that noise. For that reason, we firstly present

results using only proportional control action, to show that the damping effect of the derivative control action is definitely necessary in this case, if we want to increase the velocity and position gains to achieve faster response, eliminate overshoot and improve steady-state behaviour. Fig. 3.14 shows a positioning test using the final pressure controller along with a P-only velocity controller in cascade with a P-only position controller. Without any damping from a derivative control action, the gains must be low for the closed-loop system to be stable, resulting in poor overall performance. It should be noted, however, that the steady-state errors are very promising at first glance.

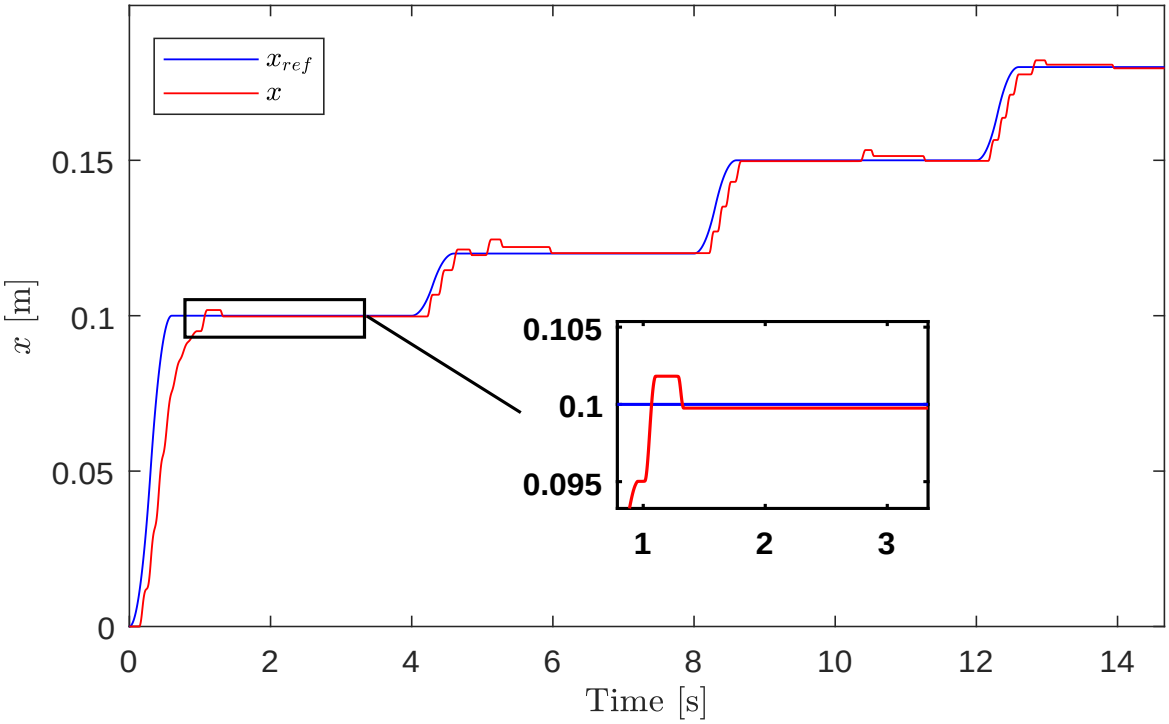

Fig. 3.14 Positioning test for P-only position and velocity controllers.

Velocity Controller: PD (filtered derivative); Position Controller: P-only

The introduction of a derivative control action allows us to increase the positioning and velocity gains (up to a point) without compromising closed-loop stability. Before presenting a generic result with the PD controller to verify the performance improvements, an important drawback of derivative action should be discussed: the differentiation of a noisy signal tends to amplify that noise, negatively influencing the overall control action.

It is therefore common-practice to filter the derivative term, so as to reduce its high-frequency gain (Åström and Hägglund, 1995). In the frequency domain, the control action of a PD controller with filtered derivative is thus given by equation 3.16:

$$U_{PDf} = KE(s)\left(1 + \frac{sT_d}{1 + sT_d/N}\right) \tag{3.16}$$

This is equivalent to filtering the derivative by a first-order low-pass filter with time constant T_d/N. The literature suggests a value between 8 and 20 for N (Åström and Hägglund, 1995).

Velocity Controller: P-D; Position Controller: P-only

Now, another way of avoiding the possible problems brought forth by using derivative action is to completely avoid differentiation - in this case, this means using the estimated acceleration value provided by the Kalman filter instead of differentiating the velocity error (which also uses an estimated value of velocity). This also helps in avoiding the problems caused by the abrupt reference change of a step input, since we are only using the feedback signal and not the reference. This avoids the introduction of a zero in the CLTF, generally resulting in less overshoot. However, in this case the acceleration is estimated by the Kalman filter based on the position measurements, which means there is also the filter's dynamic response and estimation noise to consider, making the choice between differentiation and estimation a not so obvious one. Nonetheless, and because the differentiation in the PD is done on what is already a noisy estimate provided by the Kalman filter, the implementation of a **P-D** velocity controller was considered definitely worth the try.

As was already mentioned, the P-D controller replaces the term that is proportional to the derivative of the velocity error with a term that is proportional to the direct estimate of the acceleration. After proper tuning, the results obtained with a PD and P-D velocity controllers can be seen in Fig. 3.15.

Both controllers show significant improvements in performance. At first glance, steady-state errors seem to be satisfactory and the movements between positions are smooth. Although the performance is similar, the P-D seems to present slightly better results, and despite it not being evident from Fig. 3.15, throughout testing it showed an overall smoother final approach to the desired position, with a better following of the S-curve input up until the vicinity of the final position. Since it is also the safest choice due to possible noise-amplification issues with the PD (even with filtering), and because it helps in smoothing out the response to a step-input, we proceed with the P-D controller. Recall that all results presented in this section are for S-curve inputs (the aim is to have optimal performance for those inputs), but that does not mean that the step-response should not be considered - it will be tested in Chapter 4.

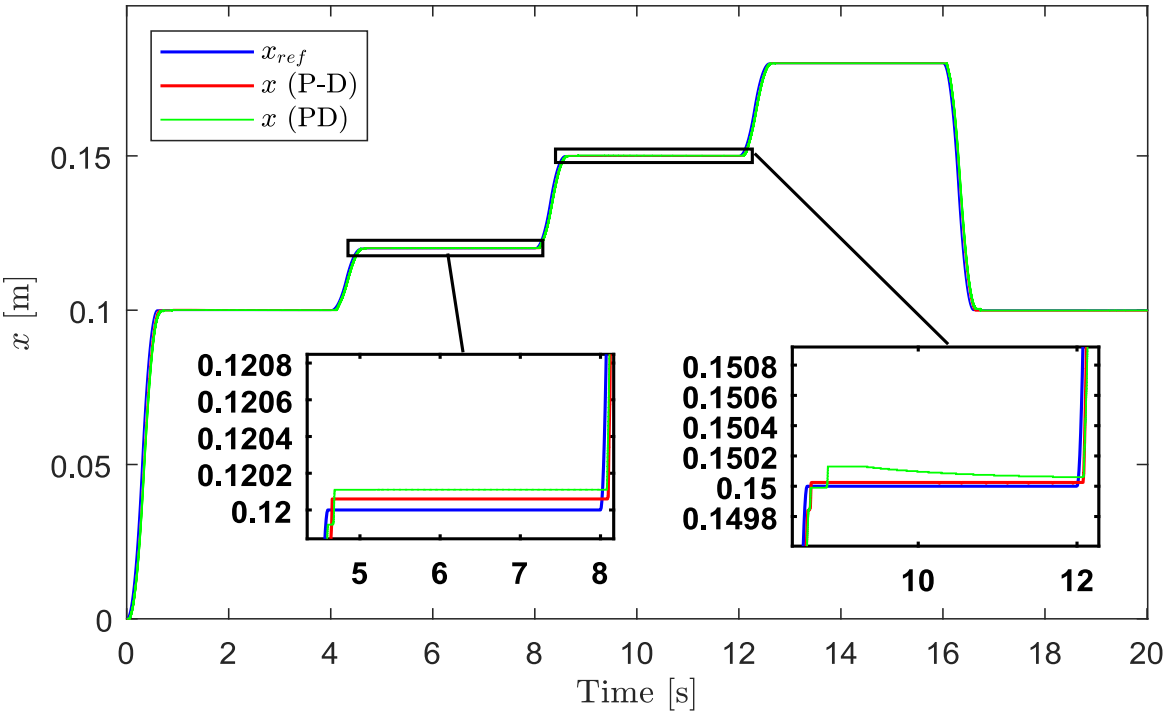

Fig. 3.15 Positioning test for P-only position controller with a PD (with filtered derivative ($N = 14$)) and P-D velocity controllers.

Velocity Controller: P-D with a pressure error ($\varepsilon_{pB} = p_{Bref} - p_B$) term

Let us now recall the coupling issue discussed in Section 2.5.5. It inspired an attempt to use a control action term that was proportional to the pressure error ($\varepsilon_{pB} = p_{Bref} - p_B$). Let us consider that p_B drops below p_{Bref} - we can see how that would transiently affect velocity, even though its steady-state value should remain essentially unchanged in choked flow conditions. A sudden drop in p_B would thus tend to temporarily increase velocity, and a control action term that would immediately respond to that pressure drop would help negate that effect. On the other hand, one may argue that this can also negatively affect motion control, because that pressure variation is often caused by the motion controller itself, doing exactly its intended job - modulating piston velocity through manipulation of p_B, changing the pressure differential between chambers to brake or accelerate the piston.

Let us then look at a comparison between the performances of the controller with and without the ε_{pB} term (Fig. 3.16).

One should not blindly extrapolate the overall effect of the term from the small sample of movements presented in Fig. 3.16, but throughout extensive testing it proved to have, on average, a small positive effect on performance after proper tuning.

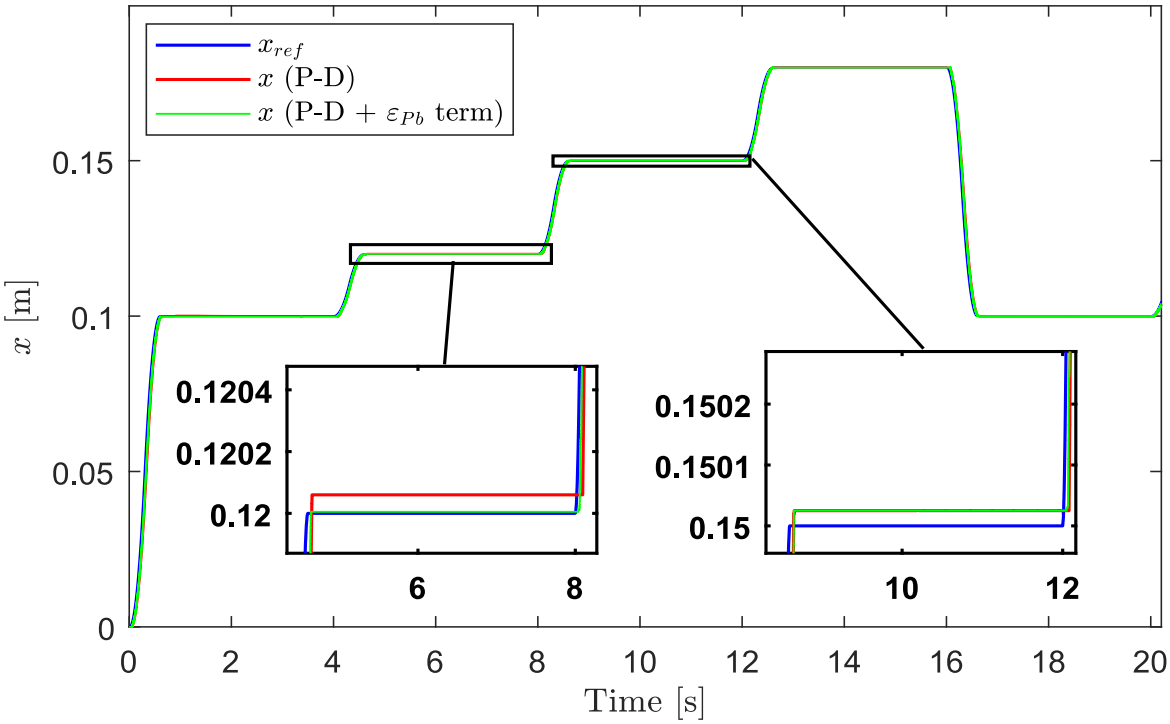

Fig. 3.16 Performance comparison for velocity controller with and without a term proportional to the pressure error in the auxiliary chamber

A major positive effect of the introduction of this term relates to an as of yet undiscussed issue: robustness. All results presented until now have featured a payload mass of $m = 4$ kg, but the ability to properly function for a range of different masses is obviously a very appealing characteristic of a controller, for most practical applications.

The ε_{pB} term proved to have a defining positive impact on the controller robustness, allowing it to perform well for a relatively wide range of masses. More thorough robustness tests will be presented in Chapter 4, but we will now provide a comparison of the performance for $m = 8$ kg, again with and without the ε_{pB} term. The movements presented in Fig. 3.17 have small amplitudes because those are the most critical, since they involve low velocities where stick-slip motion occurs, possibly leading the system into a hunting limit-cycle. Recall that the solution that was found for the limit-cycle problem (Section 3.3.1) had to be tuned, and a tuning that works for a payload mass might not work for another.

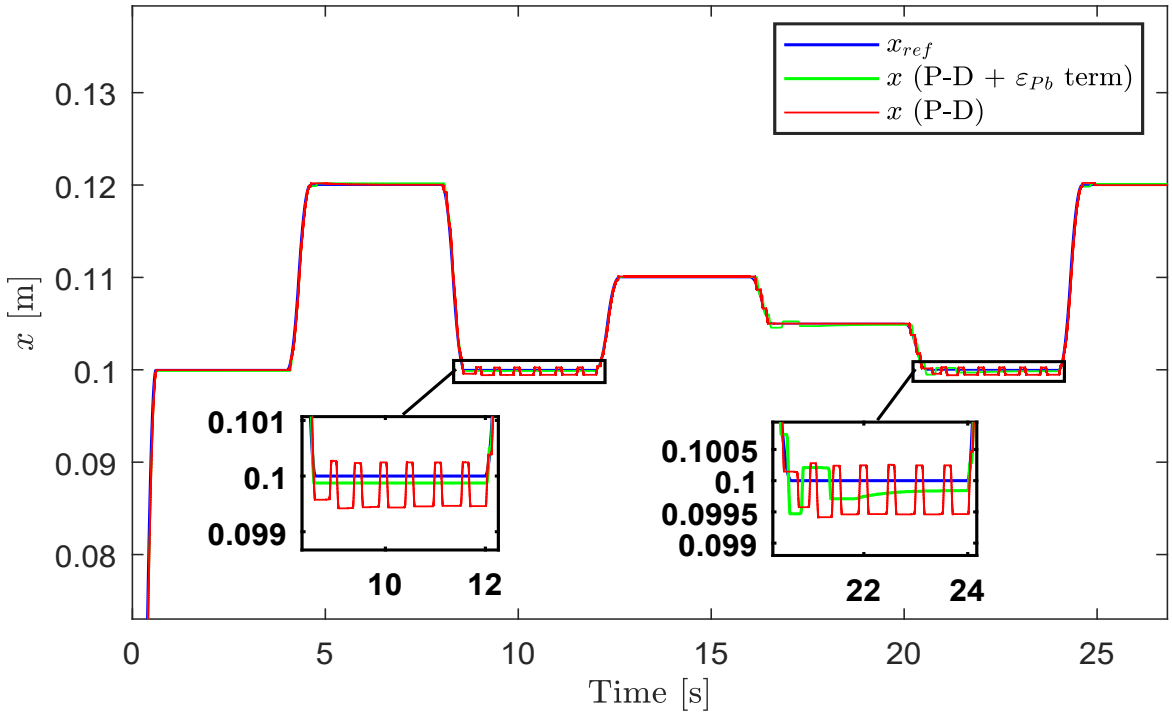

Fig. 3.17 Comparison of controller performance with and without the ε_{pB} term, for payload mass $m = 8$ kg

We can see that the controller without the ε_{pB} term is not able to guarantee stability. It is true that performance is slightly worse than for m=4 kg, but the controller with the ε_{pB} term can handle double the payload mass without significant problems.

The change in mass translates of course to a change in the inertial characteristics of the system, which will in turn alter the pressure differential needed to accelerate or brake the piston. In hindsight, the positive contribution of the ε_{pB} term to the robustness to mass changes may be justified by the fact that the controller becomes directly "aware" of the differences in the p_B equilibrium values, namely those caused by mass variations.

The control objectives now seem to have been met, from a qualitative standpoint - the steady-state errors seem to be satisfactory, the movements between positions are overall smooth and the controller is able to handle a range of payload masses. Chapter 4 will present more thorough and "standardized" results, as well as comparisons with a classical architecture PID controller ($u_A = -u_B$).

3.3.4 Controller Switching

The control architecture presented in this work foresees four different controllers: two for forward movement and two for backward movement. This is not to say that the controllers cannot handle movement in the opposite direction that they're intended for - as we'll see, small overshoot corrections are easily dealt without the need for a controller switch. Nonetheless, two switching strategies were studied, one of them monitoring the sign of the position error, where those small corrections actually involve switching controllers. The other strategy compares the new position reference with the previous one. If the position value of the new reference is higher than the previous one, the servo-valves will receive instructions from the forward controllers. If the new reference has a lower position value, then the backward controllers are activated. These strategies can be summarised by the following pseudo-code algorithms.

Strategy 1: Active controller depends on sign of position error.

```
function ActiveController1 (x_ref, x)
if (x_ref − x) ≥ 0 then
    ActiveController1=FWD ;          // Forward controllers are active
else
    ActiveController1=BWD ;          // Backward controllers are active
end
```

Strategy 2: Active controller depends on sign of $x_{ref}(t) - x_{ref}(t-1)$ (new reference - previous reference)

```
function ActiveController2 (x_ref(t), x_ref(t − 1))
if (x_ref(t) − x_ref(t − 1)) ≥ 0 then
    ActiveController2=FWD ;          // Forward controllers are active
else
    ActiveController2=BWD ;          // Backward controllers are active
end
```

It should be noted that all controllers are working at all times, but at a given time, only the controllers of a certain direction are connected to the servo-valves.

Concerning the switching itself, an effort should be made for it to happen as smoothly as possible. Ideally, a bump-less switching could be implemented that guaranteed that the control action of the new controller would be initialized at the same value of the previous one. In practice, the smooth switching is achieved by continually resetting the integrator of the inactive controller and setting a dynamic initial condition (Eq. 3.18). This way, when switching happens the resetting will stop and the controller will start up with the right control action. This solution was devised at a time when both the

velocity and the pressure controllers had integrators, so it could be implemented for all transitions, but that is presently not the case - the velocity controller has no integrator. It is therefore only possible to smooth the transition from a velocity controller to a pressure controller. For a smooth transition to happen, equation 3.17 specifies the equality that should be verified at the moment of switching.

$$u_{pressure} = u_{velocity} \Leftrightarrow u_{int} + u_P + u_{FF} + u_{pmain} = u_{velocity} \tag{3.17}$$

where, recalling the terms of the pressure controller, u_{int} is the integral term, u_P is the proportional term, u_{FF} is the velocity feed-forward term and u_{pmain} is a term proportional to the pressure in the main chamber. Since we can continually feed an updated initial condition to the pressure integrator, we have only to manipulate equation 3.17 to find what the value of that initial condition must be:

$$u_{int} = u_{velocity} - (u_P + u_{FF} + u_{pmain}) \tag{3.18}$$

First off, let us verify that the velocity to pressure transition is smooth by looking at the control action at the instant of switching. This example was obtained through simulation of the system with the controllers presented in Section 3.3.5.

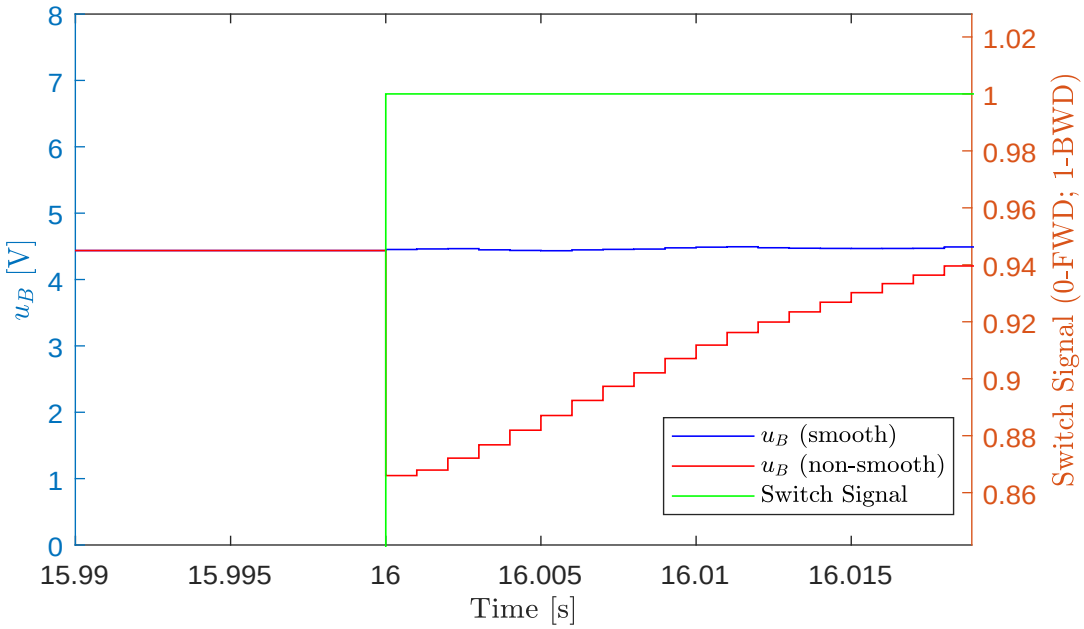

Fig. 3.18 Control action at the time of switching from velocity control to pressure control: an example.

Concerning the pressure to velocity transition, which cannot be forced to happen smoothly, it should be noted that it is not problematic at all for S-curve inputs. When the piston is stationary in a given position, the pressure control action is in its equilibrium value. When a new reference is given that requires a controller switch, the velocity controller kicks in with a proportional term, a derivative term, and the ε_{pB} term. At the instant of the switch, the system is still at equilibrium and has zero velocity and acceleration - the derivative control action is therefore null. Since the input grows smoothly (S-curve) from the previous reference to the new one, the proportional term also starts out as zero. Therefore, the overall velocity control action is initially composed only of the ε_{pB} term, which has a low gain. Fig. 3.19 shows an example of the moment of the transition for an S-curve input. We can see that there is in fact a discontinuity, but it is only a small bump. The controller is therefore able to quickly "recover" from that bump, before any negative effects have time to manifest. Negative consequences due to this small discontinuity were never observed.

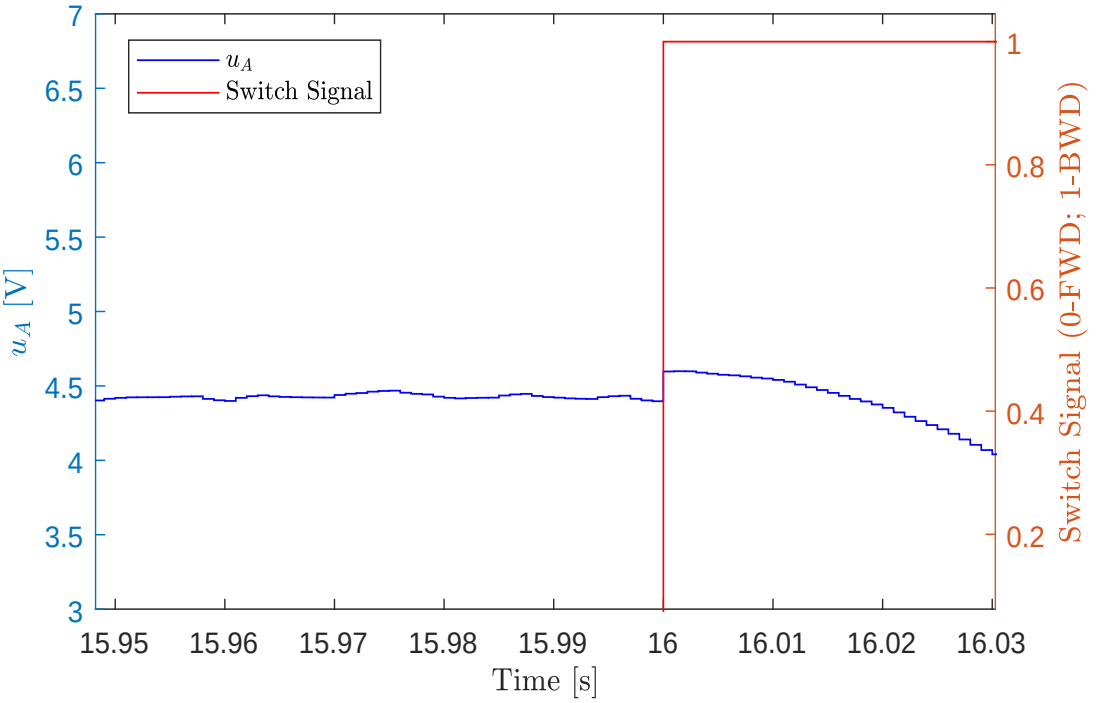

Fig. 3.19 Control action at the time of switching from pressure control to velocity control

Finally, the two switching strategies (1 and 2) are compared through an example. Strategy 1 is expected to present some problems, since the switching of the controllers for small corrections is a somewhat over-sensitive strategy. Nonetheless, both strategies were tested, and the results are shown in Fig. 3.20. Similar results were observed for different situations.

We can see that strategy 2 is clearly superior - in fact, strategy 1 seems to be virtually unusable.

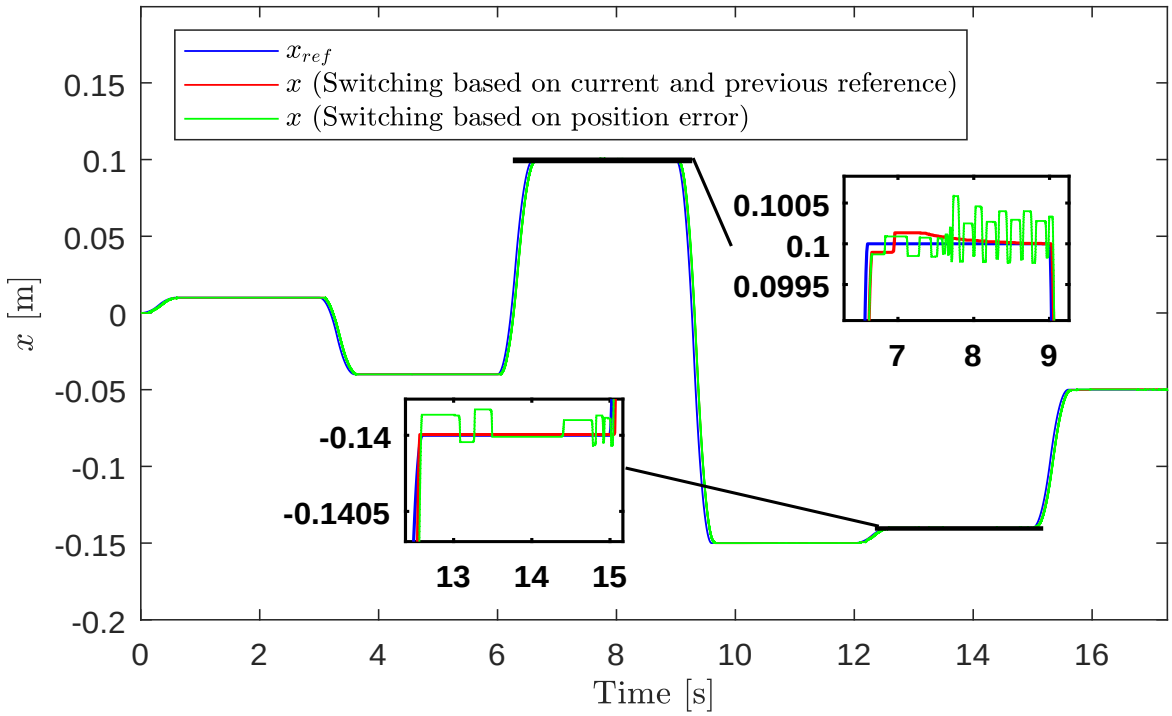

Fig. 3.20 Comparison of different controller switching strategies

3.3.5 Final Controllers

The final pressure controller consists of a PI with a $\dot{x}_{ref}$ feed-forward term, an integrator freeze-zone of ± 0.5 mm and a term that is proportional to the pressure in the main chamber (A for forward movement, B for backward movement).

The final motion controller consists of a velocity controller in cascade with a proportional position controller. The velocity control is handled by a P-D controller.

Fig. 3.21 presents a block diagram of the final overall control scheme, with both backward (BWD) and forward (FWD) controllers. Because of its size, it is shown in the next page.

The parameter values for the pressure controller were shown in Table 3.2, whereas Table 3.3 concerns the motion controller:

Table 3.3 Motion controller parameter values

Contr. Parameters	K_{pos}	K_{vel}	K_d	K_{ep}
Values	35	-40	0.5	1.45

Both forward and backward controllers function with the same parameter values. Note that for a forward movement, the velocity gain K_{vel} must be negative - a decrease in velocity is obtained by pressurising the chamber ($\uparrow u_B$), while an increase needs the servo-valve to open up to the atmosphere ($\downarrow u_B$). For a backward movement, the gain will be positive. Attention was of course paid to the signs of the control action terms for different directions.

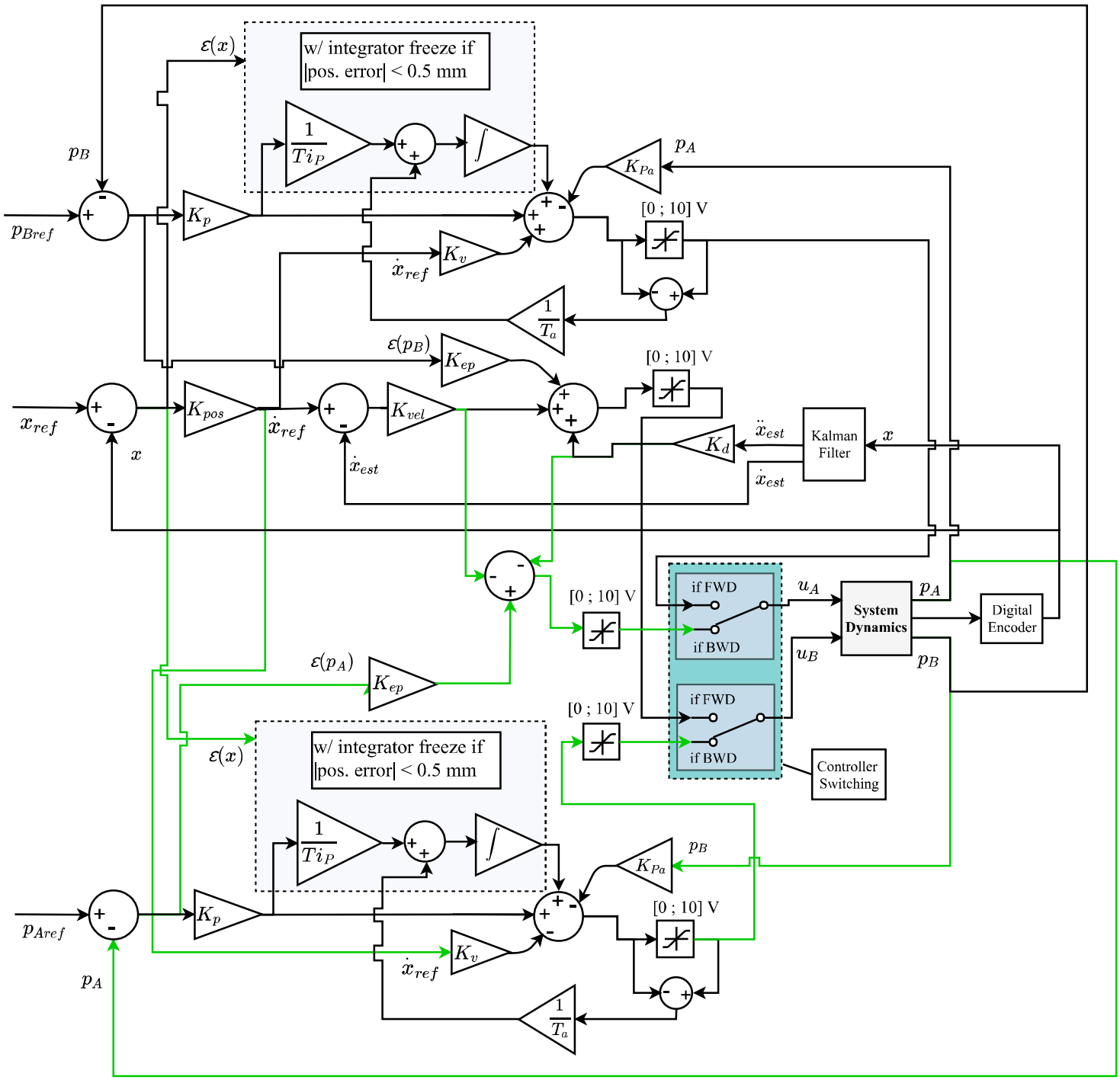

Fig. 3.21 Final control scheme. Green lines relate to the backward controller (BWD).

3.4 Chapter Conclusions

This chapter followed the control design process and discussed the switching strategy between forward and backward controllers.

The control architecture features independent motion and pressure control, with the pressure controller acting as a regulator, trying to maintain pressure in the auxiliary chamber at a constant 3 bar value. Although with the practical caveats discussed in Section 2.6.5, this allows for essentially constant choked flow through the servo-valve restriction that connects the auxiliary chamber and the atmosphere, therefore weakening the coupling between pressure and velocity. The motion controller is composed of a positioning controller in cascade with a velocity controller, i.e. the output of the first is the input of the latter.

As was discussed, the pressure and motion controllers were developed in parallel, although they were presented separately. We began with the pressure loop, but its development was discussed in the larger context of the overall objectives: accurate and smooth positioning control. Therefore, positioning results were presented and commented within the pressure-loop discussion.

The initial pressure controller comprised a PI controller with an $\dot{x}_{ref}$ velocity feed-forward term, which predicted pressure variations due to changes in the velocity reference. In order to deal with the appearance of hunting limit-cycles, a product of the interaction between integral action and friction, an integrator freeze-zone was implemented, wherein the integrator output is held constant as long as the positioning error remains lower than a pre-defined value. Having solved the limit-cycle issue, another problem was identified, which was essentially an observability problem. While the piston was stationary with constant steady-state control action u_A defined by the controller (often different from u_{A0}), p_A was still moving, unobserved, towards the "new" equilibrium value that corresponded to the current steady-state u_A. In the process of reaching this equilibrium value, p_A would often change enough to overcome static friction and cause undesired small movements after the piston had stopped. That pressure had remained unmonitored, so an additional control term was implemented, proportional to its value. The controller was then able to compensate for the pressure build-up in the main chamber, bringing the steady-state values of u_A to approximately u_{A0} and eliminating the problem.

Before presenting the motion control design process, a brief overview was provided of the Kalman filter as a velocity and acceleration estimator, making it possible for the controllers to use those otherwise inaccessible values.

The motion control design resulted in P-only positioning controller in cascade with a P-D velocity controller, with a term proportional to the pressure error in the auxiliary chamber. The P-D was compared against a PD with filtered derivative, but the P-D

proved to be the strongest candidate, since it presented slightly better performance and was the safest choice, avoiding the differentiation of a noisy signal. With this configuration, the system seemed to present satisfactory performance: reasonable steady-state errors and smooth movements between positions, for S-curve inputs. However, due to the insights gained throughout Chapter 2, an additional term was tested - a term proportional to the pressure error (ε_p) in the auxiliary chamber, with the aim of compensating for abrupt pressure variations that could transiently affect motion dynamics. When properly tuned, it had a slight positive effect on performance, but proved to be a major contribution for robustness to payload mass variations, enabling the controller to properly perform in a range of 2 to 8 kg when that was previously not possible.

Finally, the issue of controller switching was addressed. The velocity-to-pressure transition can be forced to happen smoothly, initialising the pressure integrator with a continuously updated initial condition, such that the sum of the other control actions with the integrator output matched the total velocity control action. The pressure-to-velocity transition could not be made to happen smoothly, but this did not pose a problem for S-curve inputs, as at the instant after the switch, the only non-zero velocity control action is the ε_p term, which has a small contribution.

In the end, the developed controllers seem to meet the qualitative specifications that were set: smooth motion with no overshoot and, as we'll see in the next chapter, acceptable steady-state positioning errors.

Chapter 4

Controller Performance

4.1 Introduction

This chapter will present a deeper analysis of the controller performance, where we test the step-response, S-curve response and robustness to payload mass variations, using a classical PID controller as a benchmark. We will also present a long route where we make an extensive sweep of the cylinder stroke-length with movements of varying amplitudes, to show that the controller performs well for different system properties, e.g. the large difference in the time constants of the chambers when the piston is close to one of the ends of the cylinder. Both controllers (the developed one and the classical one) are tuned to perform better for S-curve inputs, since it is a "healthier" and smoother input, and also greatly helps with the controller switching problem, as discussed in Section 3.6. That tuning was aimed for smooth movements with minimum overshoot, without too much sacrifice of steady-state positioning errors. Robustness to payload mass variations was also taken into account. It should be noted that while the developed controller handles step inputs with no major problems using the S-curve tuning, that is not the case for the classical controller, where a compromise had to be found so that some step-inputs did not lead to instabilities.

4.2 Step Response

As was already discussed, the controllers were tuned for optimal performance when inputs are given in an S-curve shape, but it is of course relevant to evaluate their step-response.

Fig. 4.1 shows the response of both controllers to a short series of step inputs.

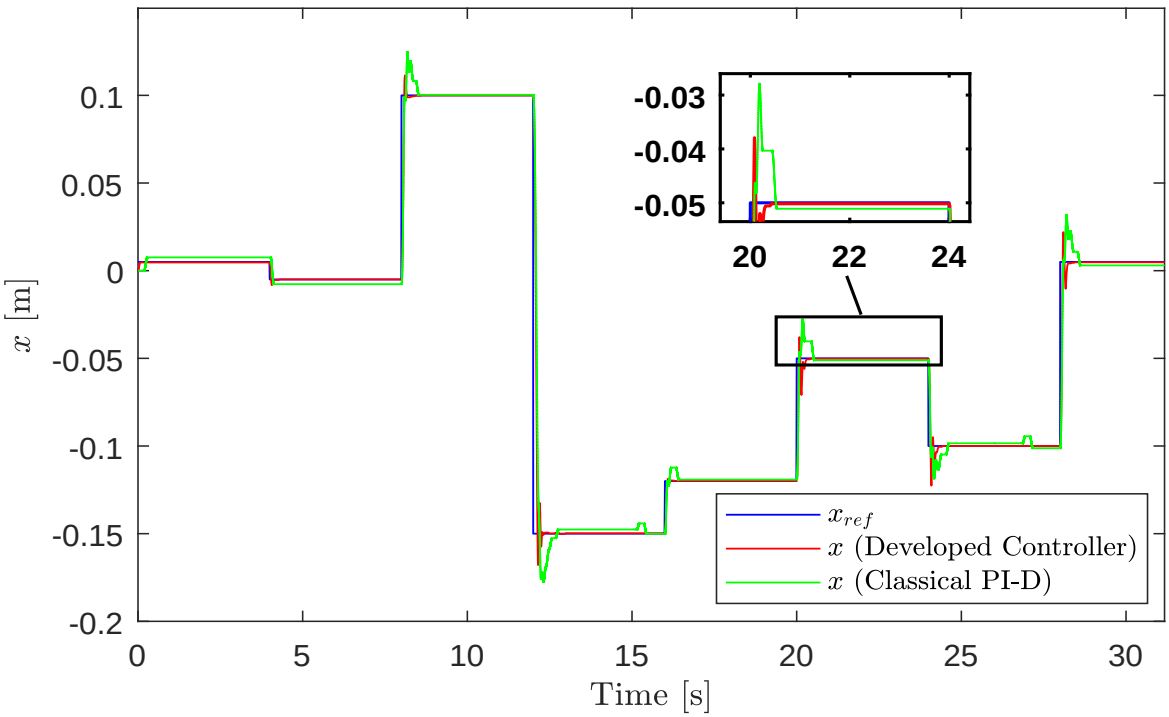

Fig. 4.1 Step-response of the developed controller and classical PI-D controller, for m=4 kg

We can see that the developed controller presents undoubtedly better performance. Both controllers show some instances of considerable overshoot, but it is usually smaller for the developed controller, even though it also shows the max. overshoot value. Its steady-state errors are also smaller, and there are no instances of the delayed small movements that happen for the PI-D, which are due to unobservable pressure dynamics in the chambers. As explained in more detail in Section 3.3.2, after the piston stops and the control action is constant, the pressures are still moving towards their equilibrium values, and this process sometimes leads to a large enough pressure differential to cause the piston to move. The performance of the controllers is more objectively quantified in Fig. 4.2 and Table 4.1.

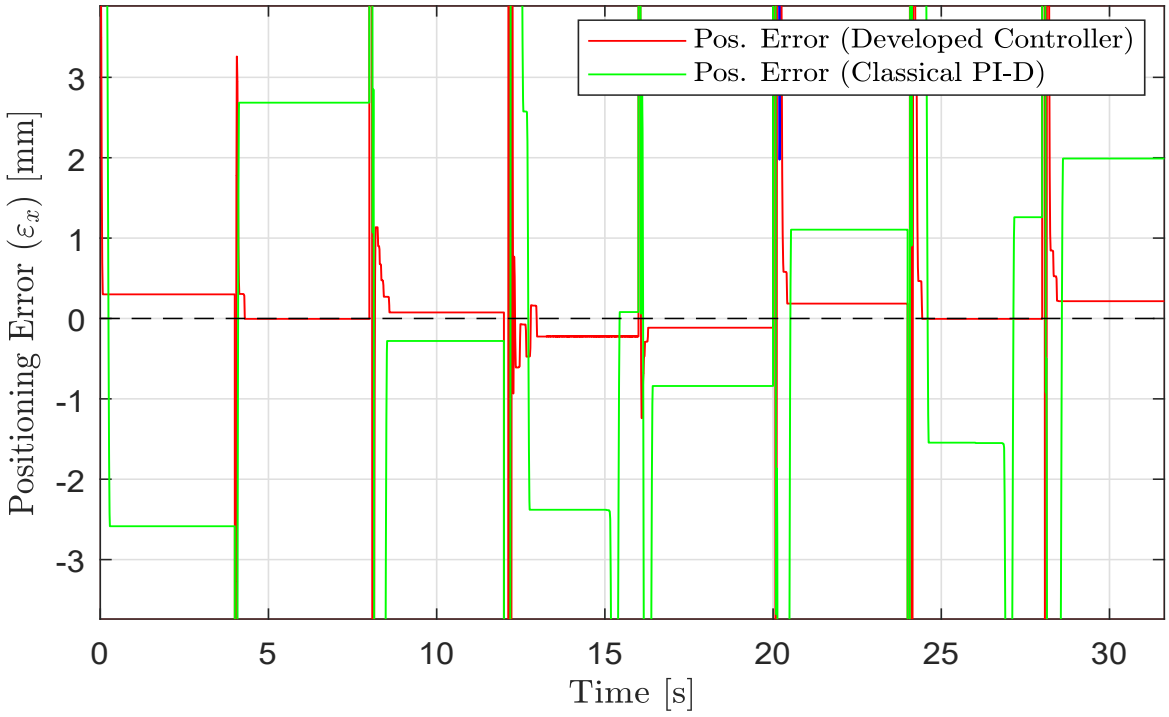

Fig. 4.2 Positioning error for the step-response test (Fig. 4.1), for m=4 kg

Table 4.1 Performance metrics for step-inputs. ε_{ss} stands for the steady-state positioning error.

Controller	Max. % Overshoot	Avg. % Overshoot	Max. ε_{ss} (mm)
Classical PI-D	36%	20.78%	2.68
Developed Controller	45%	15.32%	0.3

Even though the developed controller has a bigger max. % overshoot, it is usually lower than that of the PI-D, and the steady-state errors are often orders of magnitude better. The maximum steady-state error was 0.3 mm, which is not excellent but is quite satisfactory, especially seeing as the universal controller tuning favoured the S-curve response over the step one.

4.3 S-curve Response

An S-curve input makes for a smooth increase of the reference from its previous value to the new one. It is of course difficult to precisely track such a reference, since it is a variable acceleration curve, but we do not require excellent tracking performance - we want only for the movements to happen in an approximate S-curve shape, i.e. in a smooth fashion. It is an input that does not put a strain on the controller, helping avoid some possible problems brought forth by abrupt reference changes. So that response time is not neglected, the S-curve is such that a movement never takes more than 0.6 seconds. For the S-curve response test, we use the same reference values as for the step-response test. The results are shown in Fig. 4.3

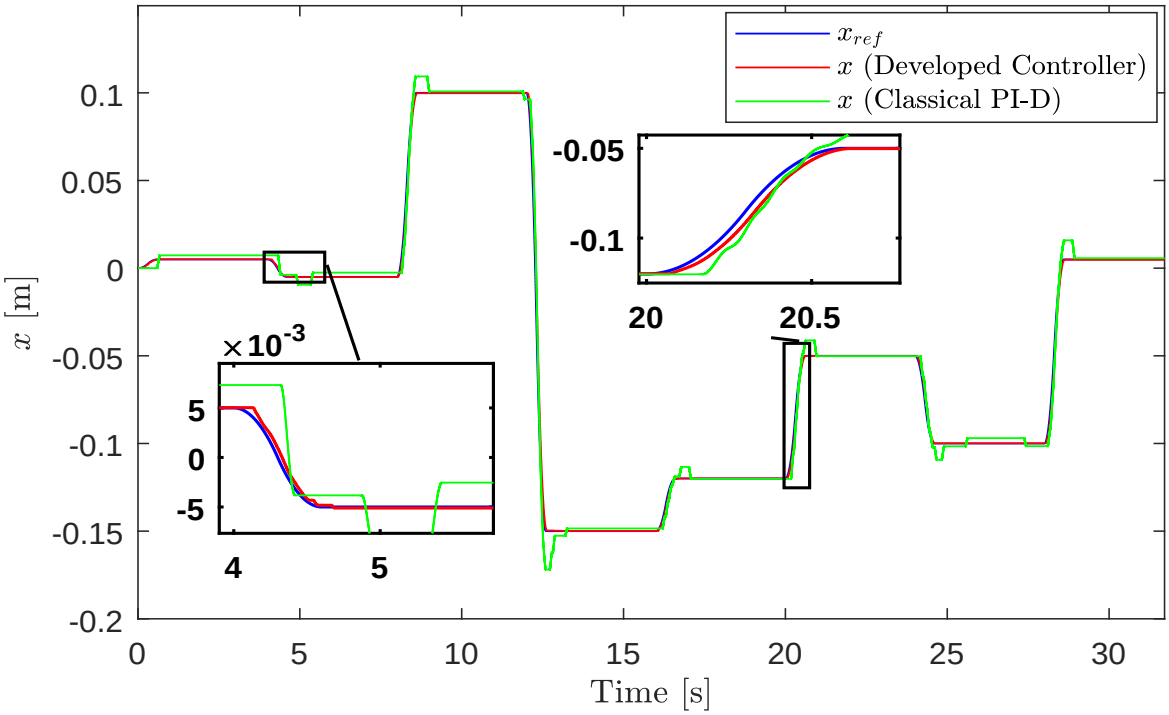

Fig. 4.3 S-curve responses of the developed controller and classical PI-D controller, for m=4 kg

Both controllers present better performance than for step inputs, as expected. Overshoot was virtually eliminated for the developed controller, and we can see that it does a good job of tracking the S-curve profile, with the movements generally occurring in one swift motion.

Fig. 4.4 and Table 4.3 provide the positioning error plot and some performance values for this test.

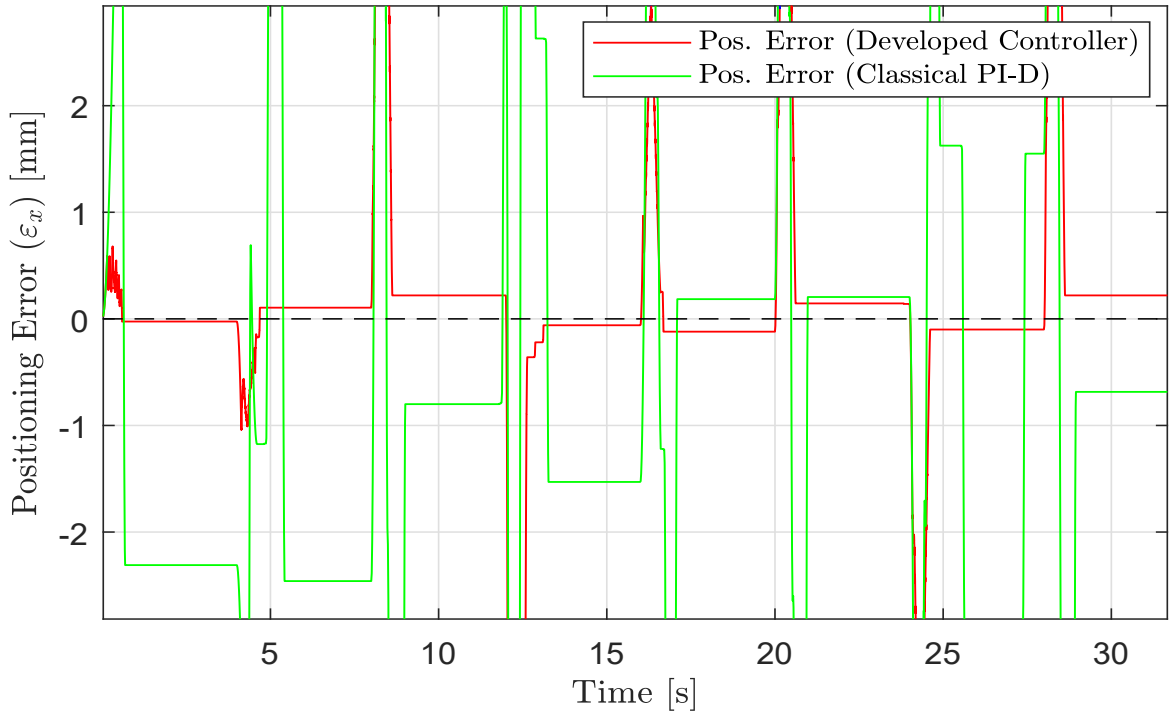

Fig. 4.4 Positioning error for the S-curve response test (Fig. 4.3), for m=4 kg

Table 4.2 Performance metrics for S-curve inputs. ε_{ss} stands for the steady-state positioning error.

Controller	Max. % Overshoot	Max. ε_{ss} (mm)
Classical PI-D	12%	2.8
Developed Controller	-	0.22

It is clearly established that the developed controller easily out-performs the classical PI-D concerning all control objectives. It presents much better overall dynamics, with smooth movements and significantly smaller steady-state errors. It must be noted, in defence of the classical PI-D, that this is not surprising - even though quite some effort was put into its tuning so that the comparison was as fair as possible, it is known that such a controller usually struggles to achieve satisfactory positioning results for typical servo-pneumatic applications. Additionally, the developed controller takes advantage of the availability of the pressure values in the chambers, which the PI-D does not.

The maximum steady-state error for the developed controller is acceptable, although not excellent, but recall that a slight compromise in tuning had to be made for the step-response's and robustness' sake. For an application where availability of S-curve

inputs was guaranteed and payload mass was constant, it is safe to assume that there would be some margin for improvement.

4.4 Robustness to payload variations

Recall that all results until now were for a constant payload mass m=4 kg. As was already briefly discussed in Chapter 3 (Control Design), robustness to variable payload mass is an interesting feature for most applications. We will present some tests with differing payload masses to establish how tolerant the controllers are to those changes.

The ε_p control action term in the velocity controller, proportional to the pressure error in the auxiliary chamber, proved to play an important role in improving robustness. The following tests will vary the payload mass in a range of [2 ; 8] kg.

For the sake of proper presentation, a table is shown here and the plots are shown in the following pages.

Table 4.3 Performance metrics for S-curve inputs, with variable payload mass m. The (*) symbol means that instability was observed.

	Controller	Max. % Overshoot	Max. ε_{ss} (mm)
m=2 kg	Classical PI-D	12.5%	2.8
	Developed Controller	-	0.22
m=6 kg	Classical PI-D (*)	18%	2.8
	Developed Controller	-	0.44
m=8 kg	Classical PI-D (*)	23%	2.8
	Developed Controller	1.7%	0.45

For a payload with m=6 kg, the classical PI-D controller is already not able to properly control the system, while we can go up until 8 kg with the developed controller with acceptable performance, although with a significant maximum steady-state error (0.45 mm).

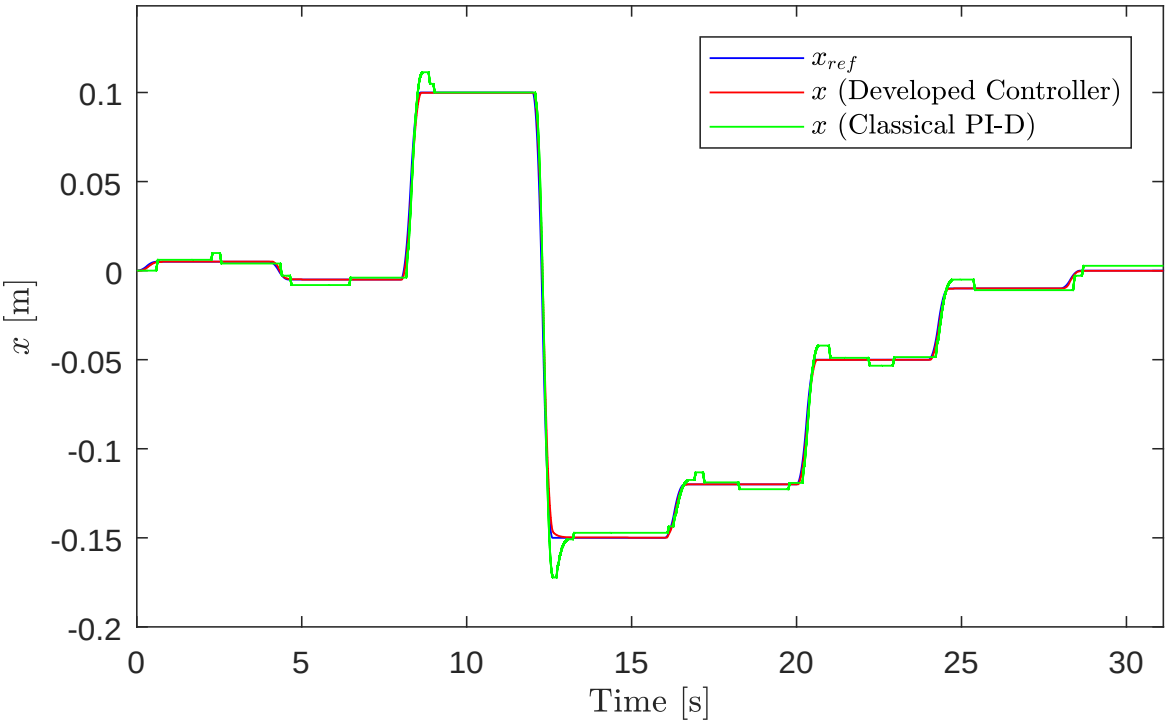

Fig. 4.5 S-curve responses for both controllers, m=**2 kg**

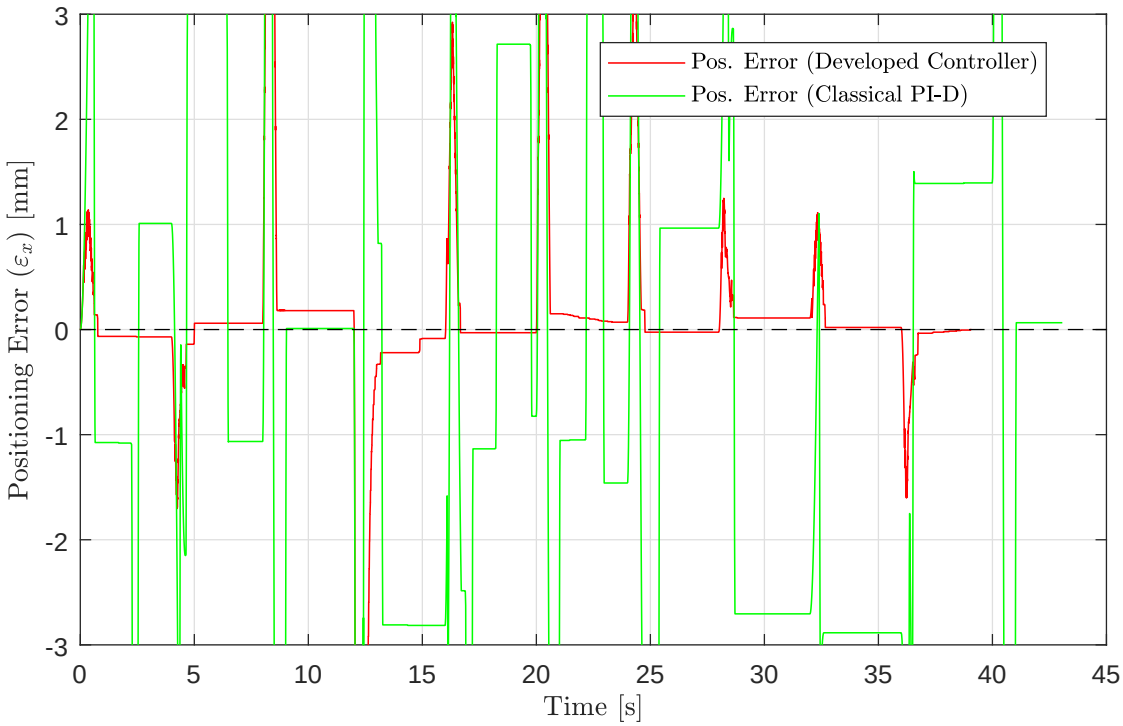

Fig. 4.6 Positioning error for the S-curve response test (Fig. 4.6), for m=**2 kg**

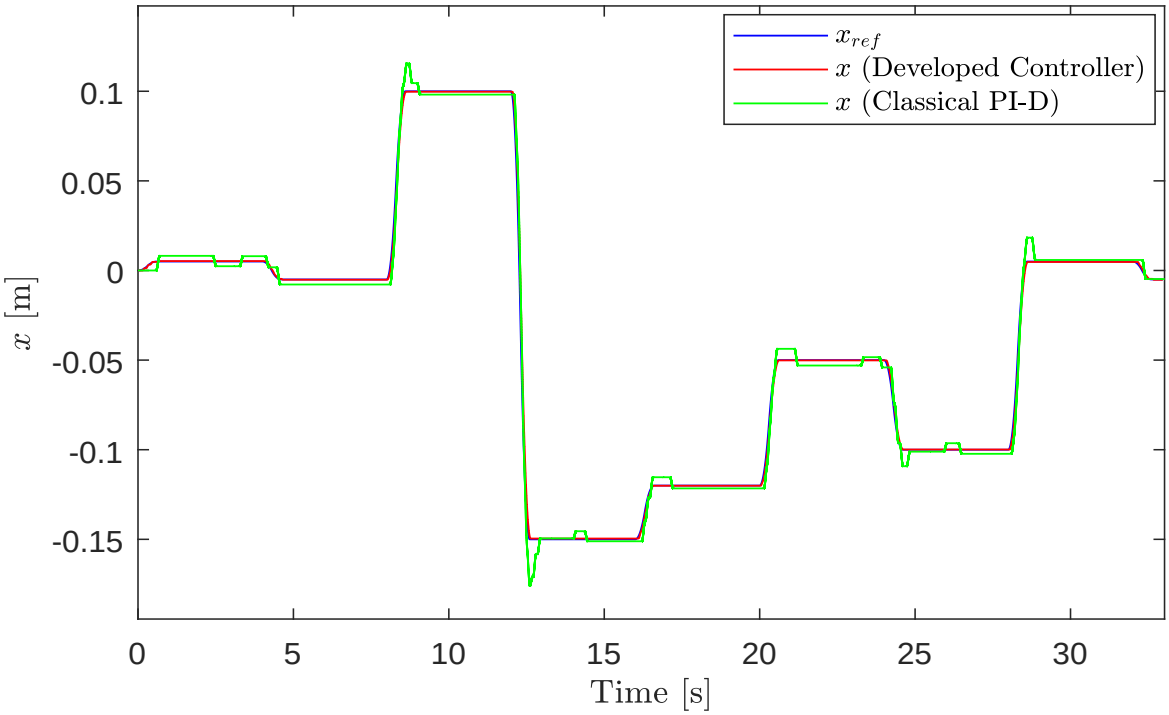

Fig. 4.7 S-curve responses for both controllers, m=**6 kg**

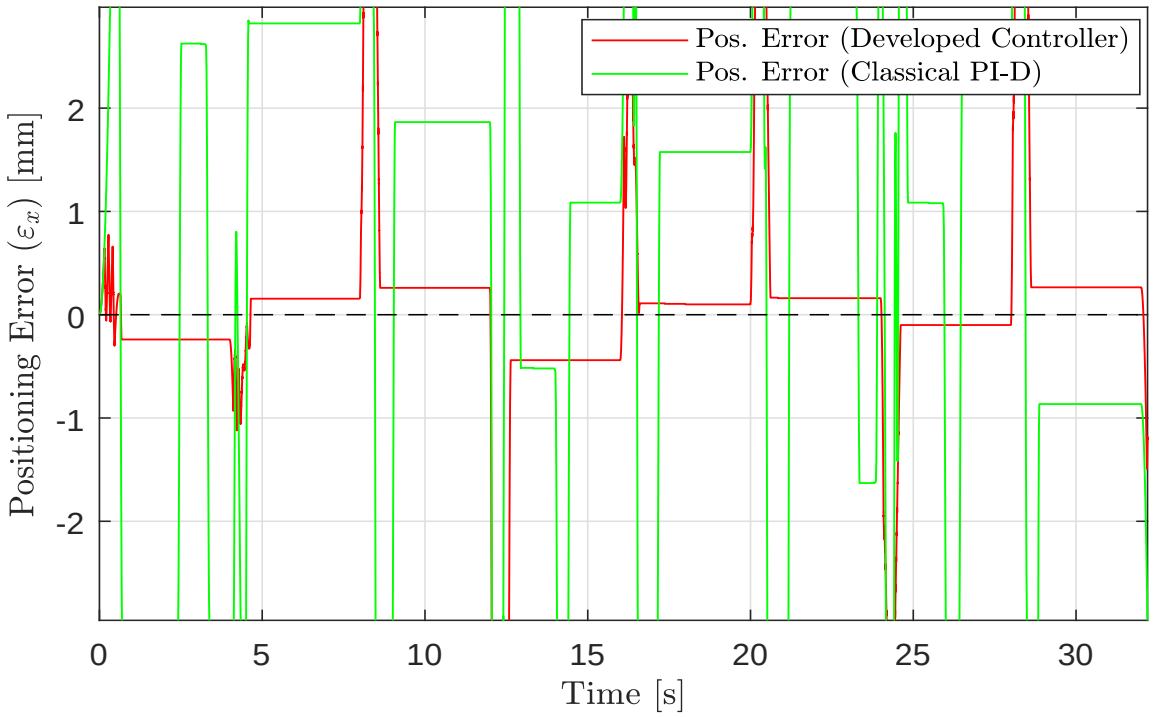

Fig. 4.8 Positioning error for the S-curve response test (Fig. 4.8), for m=**6 kg**

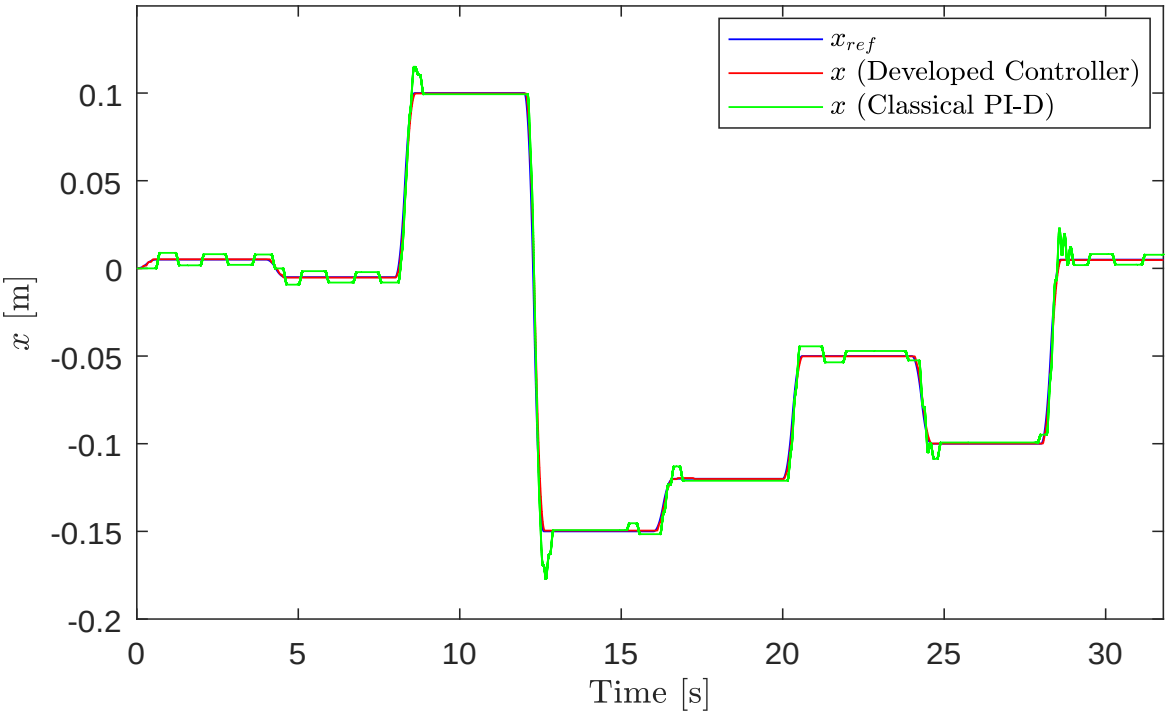

Fig. 4.9 S-curve responses for both controllers, m=**8 kg**

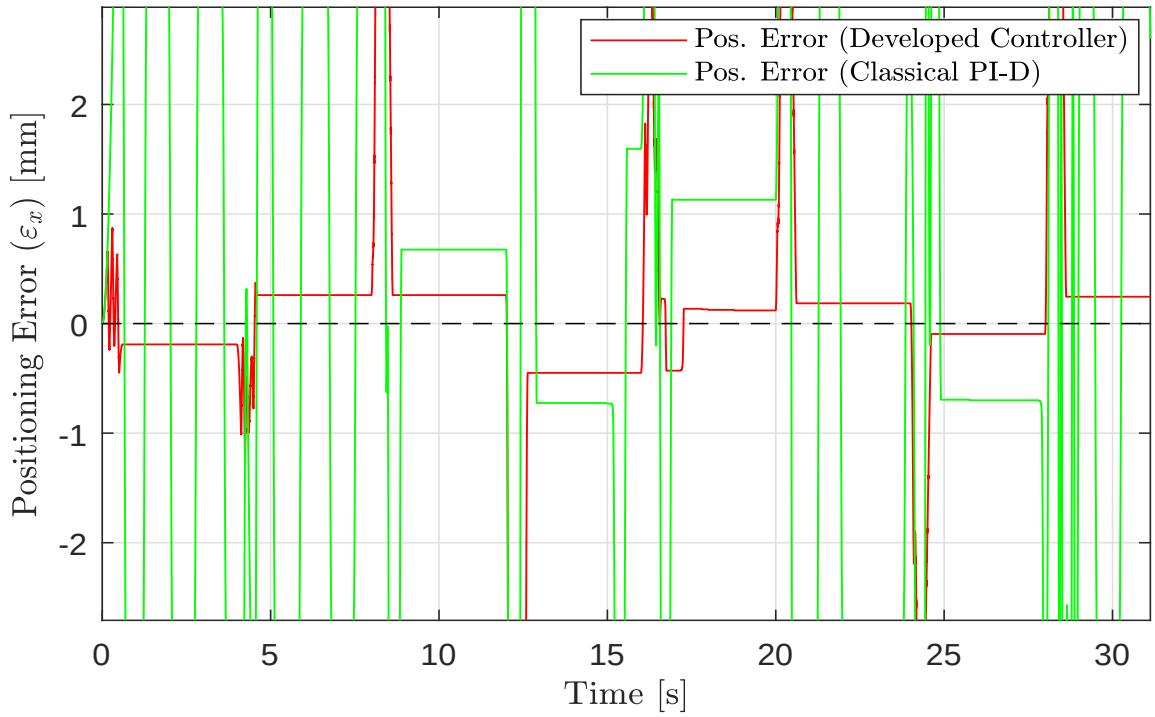

Fig. 4.10 Positioning error for the S-curve response test (Fig. 4.10), for m=**8 kg**

4.5 Path Following

It is now well established that the developed controller presents overall better performance than the classical PI-D controller. A short series of reference values was used to evaluate the step and S-curve responses, but we shall now present a more extensive test, sweeping the whole cylinder stroke-length with movements of varying amplitudes This will show that the controller is able to properly perform for small movements (where stick-slip is more of a problem) and for varying system properties, like the large difference in the time-constants of the chambers when the piston is close to the ends of the cylinder. In the beginning of the controller development process, this was a prevalent problem - the controller would sometimes function properly in the mid-area of the cylinder, but not near the ends. This test will also provide a better estimate of the true maximum steady-state error that the controller can guarantee.

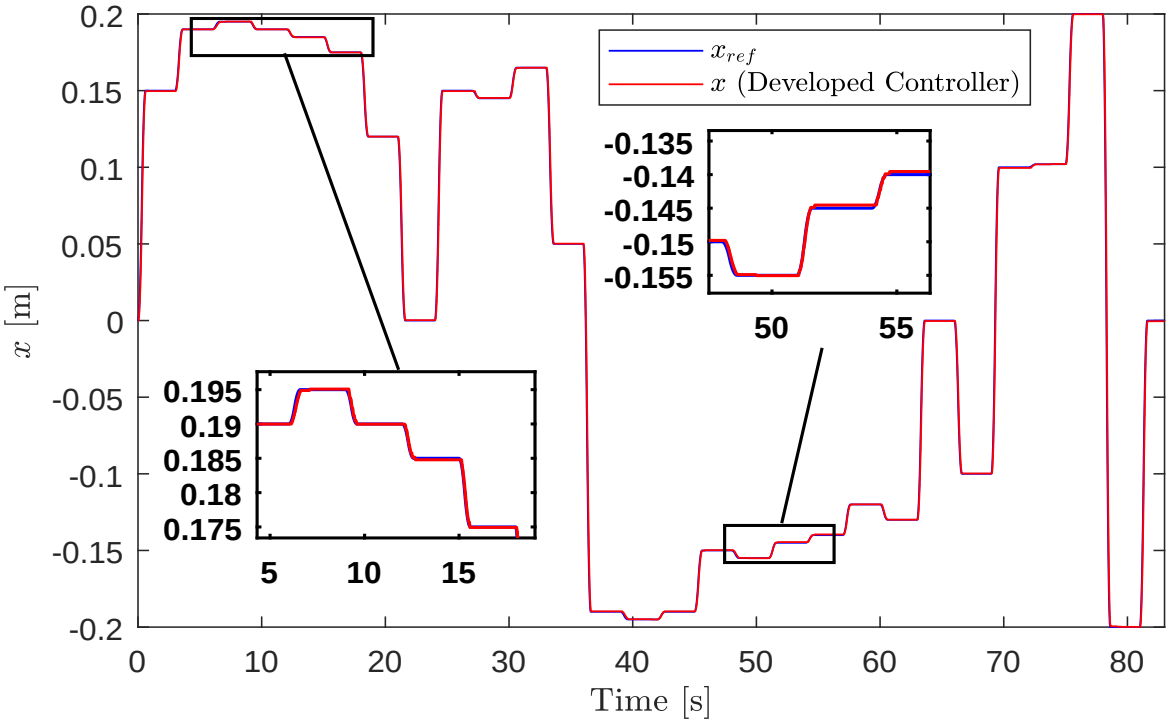

Fig. 4.11 Full sweep of the stroke length, for S-curve inputs and m=4 kg

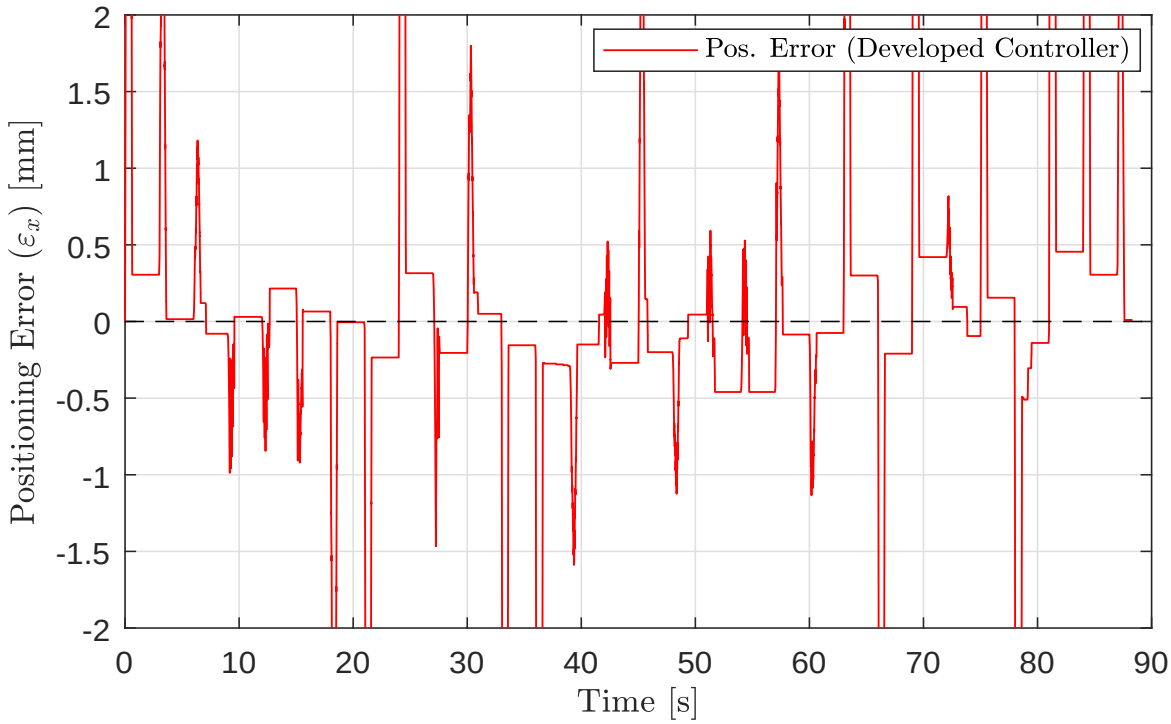

Fig. 4.12 Pos. errors for full sweep of the stroke length, for S-curve inputs and m=4 kg

We can see that the controller presents no overshoot and a shows a maximum steady-state positioning error of 0.46 mm.

It would be interesting to see what sort of positioning errors we are able to achieve with a more specific tuning of the controller, more suitable for S-curve inputs and a constant payload mass m=4 kg. That tuning is presented in Table 4.4 and the changes consist in a reduction of the integrator freeze-zone amplitude, a slight decrease in the ε_p gain, and an increase of the positioning loop gain. Figs. 4.13 and 4.14 show the path of the piston and the positioning error, respectively.

Table 4.4 Motion controller specific tuning for S-curve inputs and m=4 kg. Freeze-Zone is reduced to ±0.3 mm

Contr. parameters	K_{pos}	K_{vel}	K_d	K_{ep}
Values	42	−40	0.5	0.8

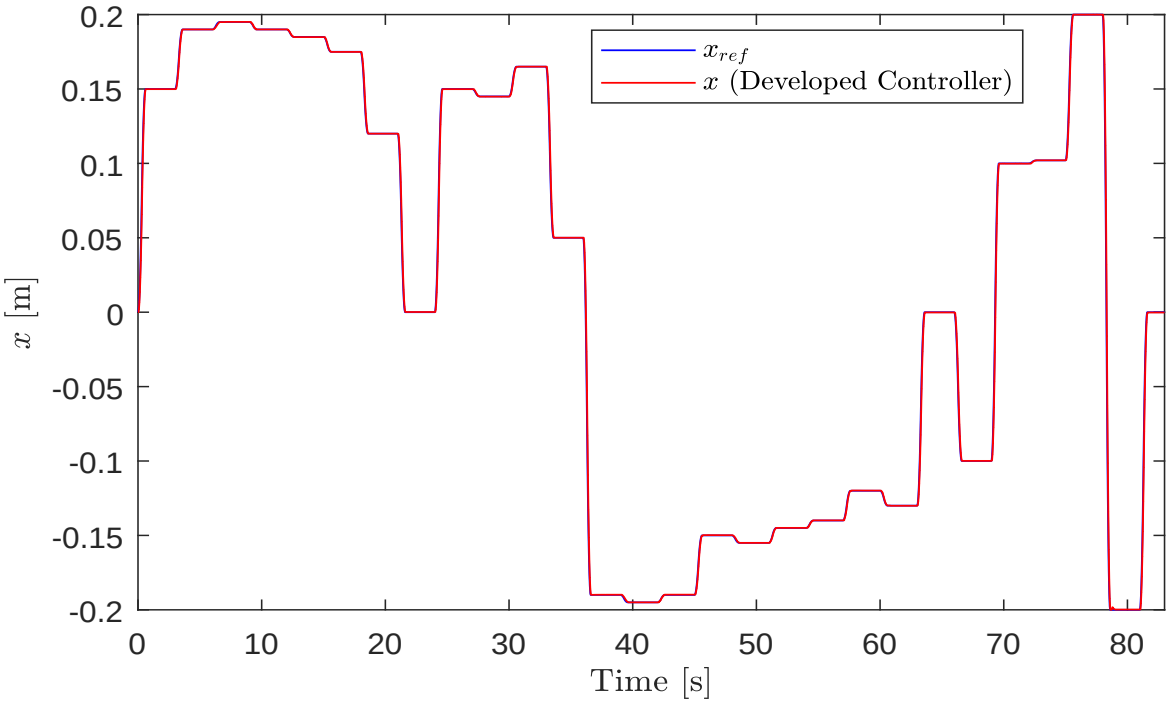

Fig. 4.13 Full sweep of the stroke length, specific tuning for S-curve inputs and m=4 kg

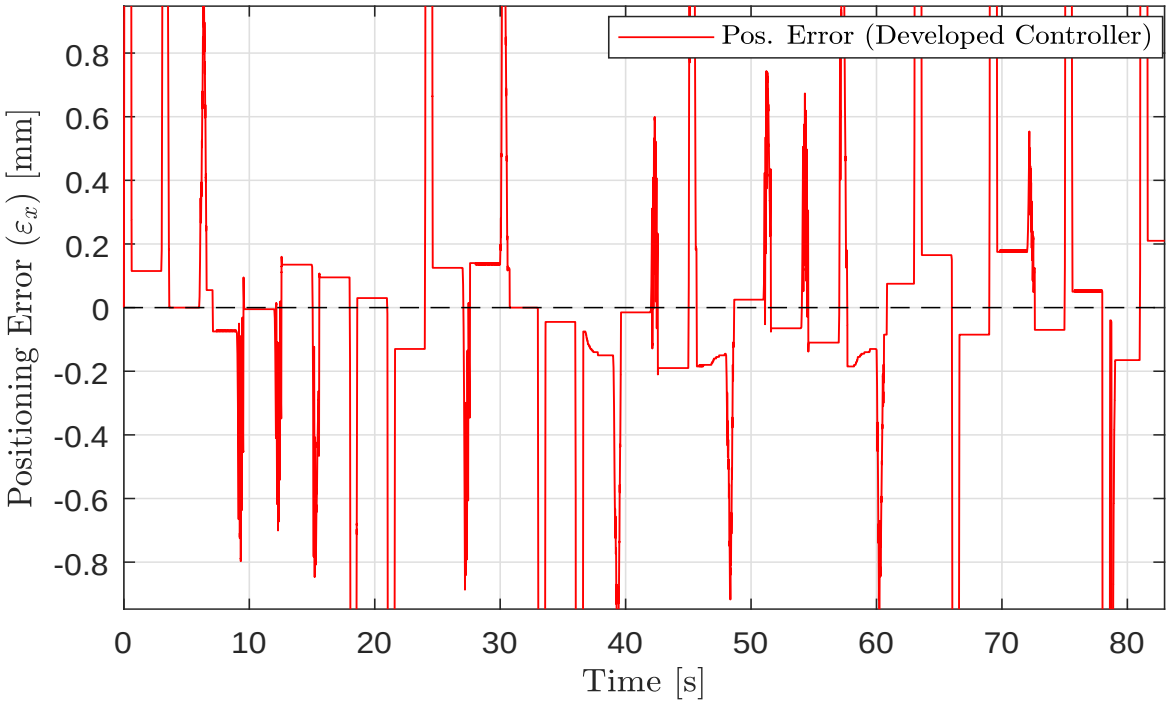

Fig. 4.14 Pos. errors for the test of Fig. 4.13, specific tuning for S-curve inputs and m=4 kg

We can see that with the specific tuning, the controller can guarantee a maximum steady state error of ε_{ss}=0.2 mm, less than half of the one obtained with the universal tuning (0.46 mm).

4.6 Discussion

On the whole, the developed controller stands quite favourably against the classical PI-D. For a step-input, it often presented smaller overshoots, even though it did have the highest max. % overshoot. The steady-state errors were significantly better.

The use of S-curve inputs was aimed at achieving smooth motion between positions, and the developed controller managed to satisfactorily track the shape of the S-curve profile, despite it being a variable acceleration curve. Most movements were done in one swift motion, with the exception of some with smaller amplitudes (typically <5 mm), where stick-slip motion was observed.

A universal tuning (Tables 3.2 and 3.3) was found, which favoured the S-curve response but allowed for acceptable step-responses. With this tuning, the controller not only presented smooth motion, virtually no overshoot and acceptable steady-state errors, it also proved robust to changes in the payload mass ([2 ; 8] kg range) a feature that is certainly interesting for most applications.

A more specific tuning was found for S-curve inputs and a constant $m = 4$ kg payload mass, which enabled a maximum steady-state error that was half of the one for the universal tuning (0.2 mm), while maintaining good dynamics.

A comparison that would perhaps be more fair, in the realm of linear controllers, would be one with a state-feedback controller. The design of such a controller (usually done by pole-placement) calls for a state-space representation of the system and assumes that feedback of all variables is available, be it through sensors or observers/estimators. It relies on the fact that if the system is completely state controllable, i.e. a control vector can bring the system from an initial state to any other state in a finite amount of time, then we can place the closed-loop poles of the system at any location by use of a state feedback gain matrix, therefore imposing the desired dynamics.

Pinto (2017), through implementation of such a controller with symmetrical control actions ($u_A = -u_B$), managed to obtain, for step-inputs, maximum steady-state positioning errors of 0.04 mm with a tuning found for maximum payload mass (higher proportional gain) and 0.17 mm for a tuning found for minimum payload mass (lower proportional gain), although with significant max. % overshoot (39% and 15%). These are undoubtedly better steady-state errors than the ones presented in this work, but they were obtained for a low-friction actuator with a Pyrex® body and a graphite piston where sealing is achieved through geometric tolerance, avoiding the friction

between conventional seals and cylinder walls, therefore minimising the problems usually associated with friction. Unfortunately, time constraints did not allow for the development of a state-feedback controller for this work.

Chapter 5

Conclusions and Future Work

At the core of this dissertation was the development of a two servo-valve control architecture for independent motion and pressure control of a servo-pneumatic system. The aim was to achieve smooth motion and accurate positioning through use of linear PID-family controllers - one for each servo-valve. The one connected to the main chamber regulates pressure in the auxiliary chamber (with a 3 bar reference value), while the other handles motion specifications. The pressure regulator helps in shielding piston velocity from being affected by pressure fluctuations, since it not only tries to keep constant pressure, it does so at a value that guarantees choked flow, therefore making the volumetric flow-rate independent of pressure. These were the general assumptions that motivated the study of this control strategy, but some work was done before initiating the control design process *per se*.

Concerning the review, update and study of the system model

We began by providing a brief state-of-the-art of positioning control in servo-pneumatics, along with the possibilities introduced by a two servo-valve architecture. We saw that SMCs are typically able to provide very accurate positioning, often presenting results in the range of tens or hundreds of micrometres, although some linear controllers have also been successfully implemented with satisfactory results.

We then proceeded to review an existing mathematical model of the system. There is a servo-valve model that provides the mass flow-rates entering or exiting the actuator chambers, and a thermodynamic model that describes the evolution of pressure and temperature in those chambers. A mechanical model describes the motion dynamics of the piston, based on the pneumatic force differential, friction and external forces acting on the system. The friction force, so important for its disrupting effects, is predicted by a LuGre friction model. The servo-valve model was updated to contemplate inverse

flow in its restrictions, in the event that the pressure in a chamber rises above p_S or falls below p_{atm}.

The aforementioned sub-systems make up a complete 6th-order non-linear model of the servo-pneumatic system. That model could be reduced to a 4th-order one (without significant loss of accuracy) by neglecting temperature dynamics, assuming the thermodynamic process inside the chambers to be polytropic. Resorting to a Taylor-series expansion limited to its first term, we linearise the model around an equilibrium point and subsequently find its coefficients. Having the 4th-order linear model, we represented it in state-space, which in turn allowed us to obtain the open-loop transfer-function representation. Seeing as the control system is a MIMO, we ended up with a transfer function matrix, whose components are the individual transfer functions between each input and each output. With this transfer function matrix, along with a block-diagram representation, it became clear how pressure and velocity are coupled. The transfer functions of interest $\left(\frac{\dot{X}(s)}{\delta U_b(s)} \text{ and } \frac{\delta p_B(s)}{\delta U_a(s)}\right)$ were presented. Since we have a MIMO system, typical SISO stability analysis tools, such as the root-locus, are not well-suited - we therefore limited ourselves to a brief qualitative analysis of the transfer functions and block diagram, which showed how the closing of one feedback loop to control a variable also influenced the other.

This coupling between pressure and velocity was then studied taking into account the normal working conditions, namely the fact that choked-flow can be assumed in both servo-valve restrictions. In practice, piston velocity is defined by the volumetric flow-rate of air entering/exiting the chamber, and as we discussed, in choked-flow conditions that volumetric flow-rate is independent of pressure - there is a natural decoupling of velocity and pressure. In normal conditions, when the piston has suffered its initial acceleration and is already moving towards its desired position, the servo-valve of the auxiliary chamber regulates the velocity in a sort of meter-out configuration, which means our restriction of interest is the one between the chamber and the atmosphere. Ideally, we would only have to consider air flow (which we can assume to be always choked) through that restriction, but in practice there is leak flow through the other one. As we saw, this leak makes piston velocity sensitive to pressure in the auxiliary chamber, which means that we cannot universally assume that choked flow in one restriction protects piston velocity from pressure variations. We can, however, study how much influence the leak flow has on this matter. The problem is that at low speeds, the leak flow is not at all negligible relative to the "main" flow - its impact is quite significant. An analysis was made where we took the partial derivative of velocity relative to pressure and evaluated it for different velocity values. It was seen that the derivative did indeed increase with decreasing velocity, although less than expected. Nonetheless, the percentage change in velocity, for a change in pressure, decreased quite rapidly with velocity (Fig. 2.13). As expected, the leak flow becomes decreasingly significant and the relative impact of pressure fluctuations is heavily diminished. It seems that save for low velocities, the effects of the coupling between pressure and

velocity are not as severe as we thought, as in practice choked-flow conditions allow for a partial decoupling of those two dynamics.

On what concerns model implementation in MATLAB/*Simulink*®, some additions were made, aimed at making the simulation environment a more accurate representation of reality. All values were updated at a 1 kHz frequency, and position and pressure signals were duly discretised in time and space. The quantization of the position readings due to encoder resolution (5 µm) was implemented, and noise was added to the pressure values to emulate the noisy pressure readings obtained in reality. For that, a real set of pressure data was examined and its variance obtained. Assuming white gaussian noise, variance and sample time (in a discrete process) are enough to characterise it and implement it in *Simulink*®. Additionally, first-order low-pass filters for the pressure readings were implemented, simulating the ones in the real experimental set-up.

Concerning control design and performance

Control design involved the development of two separate control-loops - motion and pressure. Before initiating the control design process, a Kalman filter was implemented so that the motion controller could have access to otherwise inaccessible values - it was used as a velocity and acceleration estimator.

As was already extensively discussed in Chapter 3, the development of the pressure and motion controllers was mainly done in parallel, but presented in different sections for the sake of clarity and structure. We began with the pressure control-loop, but its development was discussed in the larger context of the ultimate objective - accurate and smooth motion control - so positioning results were also shown. The building block of the pressure controller was a PI controller, to which we added a $\dot{x}_{ref}$ feed-forward term, based on what was learned in Chapter 2. In the light of the ultimate objective, two main problems were identified, originating with this initial controller:

- The presence of a hunting limit-cycle, due to the non-linear nature of friction associated with the integral control action. This was solved by implementing an integrator freeze as a function of positioning error: while the positioning error remains smaller than a pre-defined value, the output of the integrator will be held constant and equal to its value at the instant it entered the freeze-zone, which has an amplitude of ± 0.5 mm.
- The pressure creep in the main chamber while the piston was stationary, which resulted in undesired small movements. This proved to be an observability problem: since u_A could settle with a slight offset relative to u_{A0}, the equilibrium value of p_A would also differ from p_{A0}, and there would be a period of time where the piston was stopped while p_A was still moving towards its new equilibrium

value, without the controller being able to compensate for it. On the way to this new equilibrium condition, the pressure would often reach a value that was enough to overcome static friction and cause undesired movements.

The second problem was solved by introducing a term that was proportional to the pressure in the main chamber, which was previously unobserved. That pressure could rise unchecked due to the steady-state offset in u_A, so the aforementioned control action term tried to compensate for that. As we saw, this solution managed to successfully lead u_A to approximately u_{A0}, therefore eliminating the problem.

The motion control loop featured a P-only positioning controller in cascade with a velocity controller. Integral action in the velocity controller was avoided, since its interaction with friction caused problems and we did not actually require precise velocity control, as long as the movements were smooth and steady-state errors were not compromised. We therefore began by testing a P-only velocity controller, with expectedly poor results, as it was clear some damping was needed. The implementation of a PD velocity controller (with filtered derivative) showed big improvements in performance, but since it involved the differentiation of the velocity estimate, which had quite some estimation noise, a P-D controller was also tested. The P-D showed slightly better results and since it is also the safest choice because of the noise issue, it was chosen over the PD. At this time, the controller seemed to perform acceptably, with smooth motion and tracking of an S-shaped positioning input, no overshoot (in the S-curve response) and reasonable steady-state positioning errors. Based on what was learned in Chapter 2, a term proportional to the pressure error in the auxiliary chamber was tested in the velocity controller, with the aim of compensating for abrupt pressure variations that could transiently affect motion dynamics. This term had only a slight positive impact for the standard payload mass (4 kg), but what we saw was that it made the controller more robust, allowing it to properly function in a range of masses of 2 to 8 kg, which was previously not possible. By changing the mass we are affecting motion dynamics, changing the pressure differentials needed for acceleration and deceleration, and this term seems to compensate for that change, somewhat smoothing the more abrupt pressure changes that occur for heavier payloads.

With this, we finalised the control design phase and then proceeded with the study of switching strategies between forward and backward controllers. It was seen that a smooth transition could be forced for the velocity-to-pressure switch, since the pressure integrator can be initialised with a continuously updated initial condition, such that the total control action would match the velocity control action at the instant of the switch. The pressure-to-velocity transition could not be made to happen smoothly, but this did not pose a problem for S-curve inputs, as at the instant after the switch, the only non-zero control action is the low-weighted ε_p term.

With the switching issue resolved, we proceeded to more thoroughly test the final controllers against a classical PI-D controller, with symmetrical control actions and a ±3 mm dead-zone to avoid the appearance of limit-cycles.

The universal tuning found in the design phase favoured S-curve responses for the sake of smoothness, but we began by testing the step-response. While the max. % overshoot value belonged to the developed controller (45% vs 36%), the classical PI-D usually had bigger overshoot. The maximum steady-state errors were significantly better for the developed controller (0.3 mm vs 2.68 mm).

As for the S-curve response, we began by doing a small test with the same reference positions as step-response one, for comparison with the classical PI-D, and then doing a more extensive sweep of the whole cylinder stroke-length with movements of varying amplitudes. The initial test showed once again much better results for the developed controller. It presented virtually no overshoot and showed smooth movements between positions, with most of them being completed in one swift motion, with the exception of small movements (typically <5 mm) where stick-slip was more of a problem due to low velocities. Maximum steady state-errors were 0.22 mm for the developed controller vs 2.8 mm for the PI-D. A robustness test followed, where we showed the positioning results for payload masses of 2 to 8 kg (for S-curve inputs). The maximum steady-state errors were 0.22 mm for 2 kg, 0.44 mm for 6 kg and 0.45 mm for 8 kg. The classical PI-D was not able to properly control the system for 6 and 8 kg payloads.

The aforementioned full sweep of the actuator aimed to prove that the controller could perform well in all areas of the stroke-length and for smaller movements, where stick-slip might cause problems. We could also get a more accurate value for the maximum steady-state error, which unfortunately proved to be larger: 0.45 mm. By finding a less robust tuning that involved reducing the amplitude of the integrator freeze-zone, increasing the positioning gain and slightly decreasing the ε_p term gain, we were able to reduce the maximum steady-state error, for S-curve inputs and a 4 kg payload. This tuning managed to obtain a 0.2 maximum steady-state error, half of the value obtained for the universal tuning. This shows that for an application where the payload mass is constant, interesting steady-state errors can be obtained.

These are not at all extraordinary results when compared to the usual positioning accuracy of non-linear controllers like SMCs, but they are an improvement to typical PID-family results, and motion smoothness was in fact achieved, with no overshoot. A control architecture similar to the one herein presented was employed by Smaoui et al. (2005), with the exception that the controlled pressure was that of the main chamber, and the pressure reference was not constant. Motion control was achieved with a second-order HOSMC (higher-order SMC) while pressure control featured a first-order SMC. The positioning error was approximately 0.2 mm, thus similar to the best result obtained in this work.

Overall, the developed controller seems to meet the qualitative requirements that were initially set: smooth movement between positions with acceptable steady-state positioning errors. The results seem to validate the control architecture and there is almost certainly room for improvement, be it through better understanding of the coupling between pressure and velocity or through testing of other types of controllers (e.g. state-feedback controllers) within that same architecture.

Future Work

There seem to be many options to explore within the framework of the control architecture that was presented. Regarding the coupling of pressure and velocity and multi-variable control in general, there are ways to account for the coupling by introducing a static "decoupler" matrix in the controller, based on the so-called interaction indices between the controlled variables, as described by Åström et al. (2002). This technique does however present poor effectiveness if the coupling is severe. In a more practical sense, leak-less servo-valves could maximise the decoupling provided by choked-flow conditions.

Remaining in the realm of PID-family controllers, gain scheduling could be tried - the controller gain values could be made to vary as a function of position, for example. That way, changes in system properties due to the difference in chamber volumes could be somewhat accounted for.

State-feedback controllers could be implemented as still linear alternatives to PID-family controllers. The linearised model could be used for a pole-placement design, and the dynamics of the Kalman filtered could be formally introduced into the model so that its influence is accounted for. Since we have two inputs (u_A and u_B), the feedback gain matrix is not unique for a desired set of pole locations, and this freedom could be explored in our favour - usually, it is used to maximise the stability margin (Ogata, 2001).

In the vein of the already mentioned gain scheduling strategy, a fuzzy control algorithm could be implemented, be it for PID or state-feedback controllers. With fuzzy control, different sets of working conditions can be defined, along with a set of "rules" on how to best control the system in those conditions, based on knowledge about the system dynamics. The controller can then evaluate the current working conditions and decide the control output based on the rules that have been set. In the context of this work, it would be possible to adapt the controller parameters dynamically for different working conditions, ideally making the performance independent of variations in those conditions. It is basically a method to deal with model uncertainty, adding robustness without sacrificing general performance to do it (which happened in this work) - ideally, it provides optimal performance for a given working condition.

References

ASCO-Joucomatic (2011). *Double Acting High Performance Cylinders. ISO 15552- AFNOR ISO 15552- DIN ISO 15552, Series 450-453. PES OMEGA*.

Åström, K. J. and de Wit, C. C. (2008). Revisiting the lugre friction model. *IEEE Control Systems*, 28(6):101–114.

Åström, K. J. and Hägglund, T. (1995). *PID controllers: theory, design, and tuning*. Isa Research Triangle Park, NC.

Åström, K. J., Johansson, K. H., and Wang, Q. G. (2002). Design of decoupled pi controllers for two-by-two systems. *IEE Proceedings - Control Theory and Applications*, 149(1):74–81.

Bone, G. M. and Ning, S. (2007). Experimental comparison of position tracking control algorithms for pneumatic cylinder actuators. *IEEE/ASME Transactions on Mechatronics*, 12(5):557–561.

Brun, X., Sesmat, S., and Thomasset, D. (2005). Study of sticking and restarting phenomenon in electropneumatic positioning systems. *Journal of dynamic systems, measurement, and control*, 127(1):173–184.

Carneiro, J. F. and Almeida, F. G. D. (2016). On the influence of velocity and acceleration estimators on a servopneumatic system behaviour. *IEEE Access*, 4:6541–6553.

Carneiro, J. F. and de Almeida, F. G. (2013). Using two servovalves to improve pneumatic force control in industrial cylinders. *The International Journal of Advanced Manufacturing Technology*, 66(1):283–301.

Carneiro, J. F. and de Almeida, F. G. (2015). Accurate motion control of a servopneumatic system using integral sliding mode control. *The International Journal of Advanced Manufacturing Technology*, 77(9):1533–1548.

Dahl, P. R. (1968). A solid friction model. Technical report, DTIC Document.

de Wit, C. C., Olsson, H., Astrom, K. J., and Lischinsky, P. (1995). A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425.

Eichelberg, G. (1939). Some new investigations on old combustion engine problems. *Engineering*, 148:547–550.

Falcão Carneiro, J. and Gomes De Almeida, F. (2014). Micro tracking and positioning using off-the-shelf servopneumatics. *Robot. Comput.-Integr. Manuf.*, 30(3):244–255.

Falcão Carneiro, J. and Gomes de Almeida, F. (2011). *Undesired Oscillations in Pneumatic Systems*, pages 229–243. Springer Netherlands, Dordrecht.

Falcão Carneiro, J. (2007). *Modelação e Controlo de Sistemas Servopneumáticos usando Redes Neuronais Artificiais*. PhD thesis, Faculdade de Engenharia da Universidade do Porto.

Falcão Carneiro, J. and Gomes de Almeida, F. (2012a). A high-accuracy trajectory following controller for pneumatic devices. *The International Journal of Advanced Manufacturing Technology*, 61(1):253–267.

Falcão Carneiro, J. and Gomes de Almeida, F. (2012b). A macro-micro motion servopneumatic device. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 226(6):775–786.

Ferreira da Silva, M. (2015). Desenvolvimento de um observador de estado e de forças de perturbação para sistemas servopneumáticos. Master's thesis, Faculdade de Engenharia da Universidade do Porto.

FESTO (2005). *Manual of servo-valve MPYE-5-1/8-HF-010-B*.

Fok, S. and Ong, E. (1999). Position control and repeatability of a pneumatic rodless cylinder system for continuous positioning. *Robotics and Computer-Integrated Manufacturing*, 15(5):365 – 371.

Fujita, T., Sakaki, K., Makino, F., Kikuchi, T., Kagawa, T., and Kawashima, K. (2002). Accurate positioning of a pneumatic servo system with air bearings. *Proceedings of the JFPS International Symposium on Fluid Power*, 2002(5-3):693–698.

Fung, R.-F., Han, C.-F., and Ha, J.-L. (2008). Dynamic responses of the impact drive mechanism modeled by the distributed parameter system. *Applied Mathematical Modelling*, 32(9):1734 – 1743.

Girin, A., Plestan, F., Brun, X., and Glumineau, A. (2009). High-order sliding-mode controllers of an electropneumatic actuator: Application to an aeronautic benchmark. *IEEE Transactions on Control Systems Technology*, 17(3):633–645.

Hensen, R. H., Van de Molengraft, M., and Steinbuch, M. (2003). Friction induced hunting limit cycles: A comparison between the lugre and switch friction model. *Automatica*, 39(12):2131–2137.

Le, M. Q., Pham, M. T., Tavakoli, M., and Moreau, R. (2011). Sliding mode control of a pneumatic haptic teleoperation system with on/off solenoid valves. In *2011 IEEE International Conference on Robotics and Automation*, pages 874–879.

Ogata, K. (2001). *Modern Control Engineering*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 4th edition.

Olsson, H. (1996). Control systems with friction. *PhD Thesis*.

Pereira, P. R. d. O. B. (2013). Modelação da força de atrito de um servomecanismo pneumático utilizando o modelo de lugre. Master's thesis, Faculdade de Engenharia da Universidade do Porto.

Pinto, J. P. B. d. S. (2017). Desenvolvimento de controlador de movimento para cilindro pneumático de baixo atrito. Master's thesis, Faculdade de Engenharia da Universidade do Porto.

Simon, D. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience.

Slotine, J.-J. E. and Li, W. (1991). Applied nonlinear control. *NJ: Prantice-Hall, Englewood Cliffs*.

Smaoui, M., Brun, X., and Thomasset, D. (2005). A combined first and second order sliding mode approach for position and pressure control of an electropneumatic system. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 3007–3012 vol. 5.

Sorli, M. and Gastaldi, L. (2009). Thermic influence on the dynamics of pneumatic servosystems. *Journal of Dynamic Systems, Measurement, and Control*, 131(2):024501.

Wang, Y., Su, H., Harrington, K., and Fischer, G. S. (2010). Sliding mode control of piezoelectric valve regulated pneumatic actuator for MRI-compatible robotic intervention. In *ASME Dynamic Systems and Control Conference - DSCC 2010*, Cambridge, Massachusetts, USA.

White, F. M. (2011). *Fluid Mechanics*. McGraw-Hill, 7th edition.

Zhang, J.-F., Yang, C.-J., Chen, Y., Zhang, Y., and Dong, Y.-M. (2008). Modeling and control of a curved pneumatic muscle actuator for wearable elbow exoskeleton. *Mechatronics*, 18(8):448 – 457.