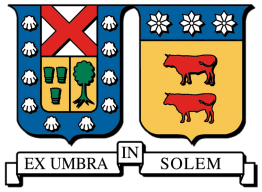
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO, CHILE



Discrete-time Approximations and Control for Port-Hamiltonian Systems

EMILIO JOSÉ OLIVARES LABRAÑA

Thesis submitted to the Electronic Department of Universidad Técnica Federico Santa María, as one of the requirements to qualify for the Bachelor's degree as **Ingeniero Civil Electrónico** and the academic Master's degree in **Ciencias de la Ingeniería Electrónica**.

Director Thesis: Dr. Héctor Ramírez (Universidad Técnica Federico Santa María)
Jury: Dr. Francisco Vargas (Universidad Técnica Federico Santa María)
Dr. Alejandro Rojas (Universidad de Concepción)

October, 2025 *Valparaíso, Chile*

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☐ Memoria o trabajo de título ☒ Tesis de Postgrado

Título del trabajo: Discrete-time Approximations and Control for Port-Hamiltonian Systems

Nombre del candidato(a): Emilio José Olivares Labraña

Carrera / Grado: Ing. Civil Electrónica Mención Control e Instrumentación - Mg. en Cs Ing Electrónica

Campus: Casa Central **Departamento:** Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Hector Ramirez Estay, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por **(marcar una opción):**

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 03-11-2025 **Firma:**

Estudiante o Candidato(a):

Fecha: 03-11-2025 **Firma:**

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Dedicado a mi mami y mi papi:
Verónica y Alexis

Agradecimientos

Durante estos años de estudio, he pasado por momentos extremadamente complejos, momentos en los cuales no me sentía útil para nada, momentos en los que quería dejar la carrera e ir a vivir como un ermitaño en el monte, momentos en los que no podía levantarme de la cama y momentos donde las ganas de vomitar estaban presente todo el día. En estos difíciles momentos siempre hubo una persona que me acompañó y apoyó sin dudarlo nunca, mi mami. Sin su apoyo incondicional estoy seguro de que no hubiera logrado ni la mitad de las cosas que he conseguido, sin ella no hubiera sido capaz de venir a estudiar a Valpo, no hubiera sido capaz de continuar cuando peor me sentía, no hubiera sido capaz de terminar mi intercambio y por último, no hubiera sido capaz de terminar la tesis este año. Ella con todo su amor incondicional, y muchas veces con su gran enojo, siempre consiguió levantarme el ánimo y hacer que continuara adelante. Por último, pero no menos importante, agradecer también a mi papi el cual igualmente estuvo presente durante gran parte de mi vida, quien, a pesar de no ser tan cercano como mi mami, es igualmente importante para mi. De él aprendí a entender a la gente y a siempre escuchar a los demás, a valorar lo que distintas personas aportan y a siempre evitar el conflicto, aunque casi nunca le hago caso. De él aprendí a hacer malos chistes y contar historias fantasiosas. Además, de él aprendí a como mantener la calma en una situación de vida o muerte como cuando se hundió el kayak en medio del mar. El único libro del colegio que recuerdo claramente es el que leí con él, La Isla del Tesoro, una historia que me leyó apasionadamente y que dejó un recuerdo muy bonito en mí. Para mi, mi mami y mi papi son todo mi mundo, en verdad los quiero muchísimo y les agradezco por ser la persona que soy ahora.

Abstract

This thesis focuses on the time-discretisation of nonlinear systems, emphasising the discretisation of port-Hamiltonian systems. Two main approaches are presented: the Discrete Gradient Method and the Collocation Method. Additionally, this work shows how to discretise controllers using Collocation Methods, as well as an implementation strategy based on the closed-loop system to be achieved, referred to as the target system strategy.

To evaluate the performance of the proposed discretisation techniques, they are applied to an electromechanical system called a piezoelectric actuator. Given its highly nonlinear hysteresis behaviour, it serves as an ideal case of study. The port-Hamiltonian model of the piezoelectric actuator used in this thesis is presented, and different discrete-time models are obtained. Furthermore, a Proportional-Integral controller capable of tracking reference signals is proposed, and a discrete-time equivalent is derived using collocation methods in conjunction with the target system strategy.

Finally, laboratory experiments are conducted to identify and validate the parameters of the proposed model using sinusoidal input tests. Subsequently, the performance is validated for pulse-type reference inputs. The discrete-time controller is then implemented on the real system, and its performance is analysed under varying sampling periods. Moreover, it is compared against a Proportional-Integral controller discretised using conventional methods.

Contents

Chapter 1

Introduction

1.1 Motivation and State of Art

In the field of automatic control and automation, system modelling plays a fundamental role. Models allow us to analyse the characteristics of systems using different mathematical tools [1–3], which in turn enable the design of control algorithms [4–7] to properly manipulate processes. Originally, control theory was developed in continuous time since controllers were implemented through physical systems. However, the rise of computing and the emergence of increasingly faster systems for control have generated the need to translate the theory into the discrete-time domain.

Regarding modelling, port-Hamiltonian system is a mathematical framework that allows us to represent multi-physical systems [8] in a formalized structure. Their main focus is the physical background of the systems and the fulfilment of the first law of thermodynamics, which ensures both the total energy conservation and the passivity of the systems [9]. This methodology enables the interconnection of systems through energy-conserving connections called *ports*, which in turn facilitate the application of passivity-based control [6, 7].

To implement computer-based control, it is necessary to connect continuous-time systems with the discrete-time framework in which processors operate. For this purpose, discrete-time approximation theory was developed for general ordinary differential equations [10]. In control engineering, the discretization of linear systems has been extensively studied, with three main approximation methods: Forward Euler, Backward Euler, and Tustin, along with the exact discretization method known as Zero Order Hold [11]. The discrete-time framework enables the implementation of advanced control techniques such as optimal control, including the Linear Quadratic Regulator, as well as Model Predictive Control and artificial intelligence-based strategies such as Reinforcement Learning, the latter two being inherently discrete-time methods [12–14]. In the context of port-Hamiltonian systems, several time-

discretization techniques have been proposed to approximate both systems and controllers while preserving their energetic properties. Two main approaches are found in the literature: discrete gradient methods and collocation methods, which have been applied for the approximation of both systems [15, 16] and controllers [17–19].

Piezoelectric actuators are electromechanical devices that enable precise position control through the application of high voltages [20]. Their modelling is particularly challenging due to pronounced hysteresis behaviour. Several approaches have been proposed to capture this phenomenon (see [21–25]), with notable contributions in [26, 27], which specifically address hysteresis modelling within the port-Hamiltonian framework. This inherent complexity makes piezoelectric actuators an excellent case study for demonstrating the methods developed in this work.

1.2 Chapter Organization

Excluding Chapters 1 and 6, which cover the introduction and conclusions, respectively, the thesis is organized into four main chapters:

- **Chapter 2:** The mathematical framework for continuous-time systems is presented. This chapter introduces mathematical models for control, linear and linearized system models, associated stability theory, port-Hamiltonian systems, and two different control techniques.
- **Chapter 3:** The concepts presented in **Chapter 2** are translated into the discrete-time domain. Time-discretization theory is presented, including methods to approximate both systems and controllers.
- **Chapter 4:** Piezoelectric actuators are introduced, and the control theory from **Chapters 2** and **3** is applied to a port-Hamiltonian system model of a piezoelectric actuator. In particular, a PI controller is proposed to stabilize the system at a desired equilibrium point. A discrete-time approximation for this controller is also presented.
- **Chapter 5:** An experimental implementation is conducted. The controller developed in **Chapter 4** is applied to a real piezoelectric bender. Two different tests are performed: the first explores the system performance under varying sampling times, and the second compares the performance of the designed controller with a classical discretized controller.

Chapter 2

Continuous-time Framework

Automatic control is a branch of electronics concerned with the regulation of physical systems through mathematical algorithms. The design of such algorithms necessitates a mathematical model of the system, which provides an abstraction of reality while capturing its essential characteristics. These models are typically expressed using differential equations that describe the system's dynamics, that is, how its behaviour evolves over time. As these phenomena unfold continuously, the system is considered to operate in continuous time.

This chapter aims to establish the mathematical framework underlying automatic control. It introduces the fundamental concepts utilised in this field and presents two distinct approaches for the design of controllers.

2.1 System Definitions

In this work we will use dynamical systems that are modelled by a finite number of coupled first-order differential equations [1]:

$$\begin{aligned}
\dot{x}_1 &= f_1(t, x_1, \ldots, x_n, u_1, \ldots, u_p) \\
\dot{x}_2 &= f_2(t, x_1, \ldots, x_n, u_1, \ldots, u_p) \\
&\vdots \quad \vdots \\
\dot{x}_n &= f_n(t, x_1, \ldots, x_n, u_1, \ldots, u_p)
\end{aligned}$$

where $\dot{x}_i$ denotes the time derivative of x_i and u_1, u_2, ..., u_p are specified input variables. We call the variables x_1, x_2, ..., x_n the state variables, they represent the memory that the dynamical system has of its past. We usually use vector notation to write these equations in

a compact form, let be

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \end{bmatrix}, \quad f(t,x,u) = \begin{bmatrix} f_1(t,x,u) \\ f_2(t,x,u) \\ \vdots \\ f_n(t,x,u) \end{bmatrix}$$

the vectors of the states, inputs and function respectively, and rewrite the n first-order differential equations as one n-dimensional first-order vector differential equation:

$$\dot{x} = f(t,x,u) \tag{2.1}$$

We call (2.1) the *state* equation and refer to x as the state vector and u as the input vector. Sometimes, another equation

$$y = g(t,x,u) \tag{2.2}$$

is associated with (2.1), thereby defining a q-dimensional output vector y that comprises variables of particular interest in the analysis of the dynamical system. We call (2.2) the output equation and we will refer to the concatenation of these two equations

$$\begin{aligned} \dot{x} &= f(t,x,u) \\ y &= g(t,x,u) \end{aligned}$$

as the state-space model or simply the state model. This works focuses on state equations without the explicit presence of the time:

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= g(x,u) \end{aligned} \tag{2.3}$$

in which case the system is said to be *autonomous* or *time-invariant*.

An important concept in dealing with the state equation is the concept of equilibrium point. A point $x = x^*$ in the state space is said to be an equilibrium point of (2.1) if it has the property that whenever the state of the system starts at x^*, it will remain at x^* for all future time. For the autonomous system (2.3), the equilibrium points are the real roots of the equation:

$$f(x^*,u^*) = 0 \tag{2.4}$$

for any arbitrary input u^*. An equilibrium point could be isolated; that is, there are no other equilibrium points in its vicinity, or there could be a continuum of equilibrium points.

Example 2.1.1. *A simple pendulum is modelled by the system equation:*

$$\begin{aligned} \dot{\theta} &= \omega \\ \dot{\omega} &= \frac{-mgl}{J}\sin(\theta) - \frac{c}{J}\omega + \frac{1}{J}\tau_{ext} \\ y &= \theta \end{aligned} \tag{2.5}$$

where $x = \begin{bmatrix}\theta & \omega\end{bmatrix}^T$ *are the system states and* $u = \tau_{ext}$ *is the system input. Here,* m *is the mass of the pendulum,* l *is the distance from the pivot to the center of mass,* g *is the acceleration due to gravity,* J *is the moment of inertia of the pendulum about the pivot, and* c *is the viscous damping coefficient. Equilibrium points of the system for* $\tau_{ext} = 0$ *can be computed by setting* $\dot{\theta} = \dot{\omega} = 0$*:*

$$\begin{aligned} &\textit{Equilibrium Point 1:} && \omega^* = 0 \\ & && \theta^* = 0 \\ &\textit{Equilibrium Point 2:} && \omega^* = 0 \\ & && \theta^* = \pi \end{aligned} \tag{2.6}$$

2.2 Linear Systems and Linearization

An important subclass of systems are linear systems. These systems have been extensively studied and are highly useful for controller design [3–5]. In state-space form, they are described by the following linear structure:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{2.7}$$

where A a $n \times n$ matrix that represents the interaction among the different states of the system, B is a $n \times p$ matrix that shows how the inputs interact with the states, C a $q \times n$ matrix representing the relationship between the q outputs and the system states and D a $q \times p$ matrix describing the direct relationship between outputs and inputs.

A linear representation around an equilibrium point (2.4) of the general system (2.3) can be obtained by applying a first-order Taylor series expansion, resulting in the system

matrices:

$$\begin{aligned} A &= \left.\frac{df(x,u)}{dx}\right|_{(x^*,u^*)}, \qquad B = \left.\frac{df(x,u)}{du}\right|_{(x^*,u^*)} \\ C &= \left.\frac{dg(x,u)}{dx}\right|_{(x^*,u^*)}, \qquad D = \left.\frac{dg(x,u)}{du}\right|_{(x^*,u^*)} \end{aligned} \tag{2.8}$$

Thus, the linearized model around the operating point is given by:

$$\begin{aligned} \Delta\dot{x} &= A\Delta x + B\Delta u \\ \Delta y &= C\Delta x + D\Delta u \end{aligned} \tag{2.9}$$

where $\Delta x = (x - x^*)$ and $\Delta u = (u - u^*)$ the deviations of the state and the input from the operating point. It should be noted that the linear model is only valid in a neighbourhood around the operating point, as the system moves further away, the approximation will increasingly differ from the original nonlinear model.

Example 2.2.1. *Let's consider the pendulum system* (2.5)*. A linearization around the equilibrium points* (2.6) *can be obtained using* (2.8)*:*

- *Equilibrium Point 1:*

$$\begin{aligned} \begin{bmatrix} \Delta\dot{\theta} \\ \Delta\dot{\omega} \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\frac{mgl}{J} & -\frac{c}{J} \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta\omega \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} \Delta\tau_{ext} \\ \Delta y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta\omega \end{bmatrix} \end{aligned}$$

- *Equilibrium Point 2:*

$$\begin{aligned} \begin{bmatrix} \Delta\dot{\theta} \\ \Delta\dot{\omega} \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ \frac{mgl}{J} & -\frac{c}{J} \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta\omega \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} \Delta\tau_{ext} \\ \Delta y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta\omega \end{bmatrix} \end{aligned} \tag{2.10}$$

Example 2.2.2. *An RLC circuit is a linear system represented by the following system of differential equations:*

$$\begin{aligned} \dot{i} &= -\frac{R}{L}i - \frac{1}{L}v + \frac{1}{L}v_{in}, \\ \dot{v} &= \frac{1}{C}v \end{aligned} \tag{2.11}$$

The system can be rewritten as a state-space linear system:

$$\begin{aligned} \begin{bmatrix} \dot{i} \\ \dot{v} \end{bmatrix} &= \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} \begin{bmatrix} i \\ v \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} v_{in}, \\ y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ v \end{bmatrix}. \end{aligned} \tag{2.12}$$

Shifting the order of the state variables, the system becomes:

$$\begin{aligned} \begin{bmatrix} \dot{v} \\ \dot{i} \end{bmatrix} &= \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} v_{in}, \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix}. \end{aligned}$$

Upon linearisation, different systems may exhibit similar structures, which can result in the loss of some of their physical characteristics. However, this similarity also facilitates the application of common analytical tools, enabling a systematic study and comparison of diverse systems within a unified framework.

An important feature of linear systems is that they can be analysed in the frequency domain by applying the Laplace transform [3], defined as

$$F(s) = \int_0^\infty f(t) e^{-st}\, dt, \tag{2.13}$$

where s is a complex variable. When applied to linear systems, the Laplace transform allows the derivation of the transfer function (TF) between the input and the output. For the system (2.7), its transfer function is given by

$$G(s) = \frac{F(s)}{E(s)} = C\left(sI - A\right)^{-1} B + D, \tag{2.14}$$

where $F(s)$ and $E(s)$ are polynomials in s.

A fundamental property of transfer functions is their poles and zeros, which correspond to the roots of $E(s)$ and $F(s)$, respectively. In particular, the poles are directly related to the eigenvalues of the matrix A.

Example 2.2.3. *The transfer function of the RLC circuit given in* (2.12) *is obtained by applying* (2.14)*, resulting in*

$$G(s) = \frac{\frac{1}{LC}}{s^2 + \frac{R}{L}s + \frac{1}{LC}}. \tag{2.15}$$

2.3 Stability Theory and Analysis for Systems

System stability is the ability of a system to maintain or return to an equilibrium point after a disturbance. If a system is stable for an equilibrium point (2.4), it will always converge to that specific point, otherwise, the system will diverge to infinity or to others equilibrium points. Following [1], consider the autonomous system

$$\dot{x} = f(x) \tag{2.16}$$

where $f : \mathbf{D} \to \mathbf{R}^n$ is a locally Lipschitz map from domain $\mathbf{D} \subset \mathbf{R}^n$ into $\mathbf{R}^n$. Suppose $x^* \in \mathbf{D}$ is an equilibrium point of (2.16). Our goal is to characterize and study the stability of x^*, we will always assume that $f(x)$ satisfies $f(0) = 0$ and study the stability of the origin $x = 0$.

Definition 2.3.1. *The equilibrium point $x^* = 0$ of (2.16) is*

- ***Stable** if, for each $\epsilon > 0$, there exists a $\delta > 0$ such that*

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \epsilon, \quad \forall t \geq 0$$

- ***Unstable** if not stable.*
- ***Asymptotically stable** if it is stable and δ can be chosen such that*

$$\|x(0)\| < \delta \Rightarrow \lim_{t \to \infty} x(t) = 0$$

Once stability is defined, it is necessary to find a method to determine the stability of a system. For this purpose, the following theorem will be used:

Theorem 2.3.1 (**Lyapunov Stability Theorem 1**). *Let $x^* = 0$ be an equilibrium point for (2.16) and $\mathbf{D} \subset \mathbf{R}^n$ be a domain containing $x^* = 0$. Let $\mathrm{V} : \mathbf{D} \to \mathbf{R}$ be a continuously differentiable function such that*

$$V(0) = 0 \quad \textit{and} \quad V(x) > 0 \quad \textit{in} \quad \mathbf{D} - \{0\} \tag{2.17}$$

$$\dot{V}(x) \leq 0 \quad \textit{in} \quad \mathbf{D} \tag{2.18}$$

Then, $x^* = 0$ is stable. Moreover, if

$$\dot{V}(x) < 0 \quad \textit{in} \quad \mathbf{D} \setminus \{0\} \tag{2.19}$$

then $x^* = 0$ is asymptotically stable.

A second stability theorem is developed using the linearization technique:

Theorem 2.3.2 (**Lyapunov Stability Theorem 2**). *Let $x = 0$ be an equilibrium point for the nonlinear system*

$$\dot{x} = f(x)$$

where $f : \mathbf{D} \to \mathbf{R}^n$ is continuously differentiable and $\mathbf{D}$ is a neighbourhood of the origin. Let

$$A = \left.\frac{df(x)}{dx}\right|_{x=x^*}$$

Then,

1. *The origin is asymptotically stable if $Re\{\lambda_i\} < 0$ for all eigenvalues of A.*
2. *The origin is unstable if $Re\{\lambda_i\} \geq 0$ for one or more eigenvalues of A.*

Finally, a stability theorem for transfer functions is presented [3]:

Theorem 2.3.3. *Let*

$$G(s) = \frac{F(s)}{E(s)}$$

denote the transfer function of a system. Then, the system is:

- ***Stable** if all roots of the polynomial $E(s)$ have negative real parts.*
- ***Unstable** otherwise.*

Remark 1. *The analysis presented is also valid for systems with external inputs and/or equilibrium points different from the origin, since at the equilibrium point the external input is a constant that can be treated as another parameter of the system, and it is always possible to perform a coordinate transformation that places the analyzed point at the origin.*

Example 2.3.4. *Let's consider the pendulum system* (2.5)*. We analyse the stability of the Equilibrium Point 1 using Theorem 2.3.1 setting $\tau_{ext} = 0$. A suitable Lyapunov candidate function is the total energy of the pendulum:*

$$V(\theta, \omega) = \frac{1}{2} J\omega^2 + mgl(1 - \cos(\theta)),$$

where the first term represents the kinetic energy and the second term the potential energy. The function $V(\theta, \omega)$ is positive definite around $(0, 0)$ and $V(0, 0) = 0$. The time derivative of V along the system trajectories (2.5) *is:*

$$\dot{V}(\theta, \omega) = \frac{\partial V}{\partial \theta}\dot{\theta} + \frac{\partial V}{\partial \omega}\dot{\omega} = mgl\sin(\theta)\,\omega + J\omega\left(-\frac{mgl}{J}\sin(\theta) - \frac{c}{J}\omega\right) = -c\,\omega^2.$$

Since $c > 0$, we have $\dot{V} \leq 0$, showing that V is non-increasing along the system trajectories. Finally, the system is stable for Equilibrium Point 1.

Example 2.3.5. *Let's consider the linearised pendulum system around Equilibrium Point 2, given by* (2.10)*. The eigenvalues of the system are calculated with*

$$\lambda^2 + \frac{c}{J}\lambda - \frac{mgl}{J} = 0.$$

Solving for λ*, we obtain*

$$\lambda = -\frac{c}{2J} \pm \sqrt{\left(\frac{c}{2J}\right)^2 + \frac{mgl}{J}}.$$

As one of the roots has a positive real part, the linearised system is unstable. Finally by Theorem 2.3.2, the Equilibrium Point 2 of the nonlinear system is unstable.

Example 2.3.6.

2.4 Input-State-Output Port-Hamiltonian Systems

As defined in [28], an input-state-output port-Hamiltonian system (pHs) with n-dimensional state space manifold $\mathcal{X}$, input and output spaces $U = Y = \mathbf{R}^m$, and Hamiltonian $H : \mathcal{X} \to \mathbf{R}$, is given as

$$\Sigma : \begin{cases} \dot{x} = [J(x) - R(x)]\nabla H(x) + G(x)u \\ y = G^T(x)\nabla H(x) \end{cases} \tag{2.20}$$

where $\nabla H(x)$ denotes the column vector of partial derivatives of H, the $n \times n$ matrices $J(x)$, $R(x)$ satisfy $J(x) = -J^T(x)$ and $R(x) = R(x)^T \geq 0$. The Hamiltonian $H(x)$ is equal to the *total stored* energy of the system, while $u^T y$ is the externally supplied power. In the definition of a pHs, two geometric structures on the state space $\mathcal{X}$ play a role: the internal *interconnection* structure given by $J(x)$, which by skew-symmetry is *power-conserving*, and a *resistive* structure given by $R(x)$, which by non-negativity is responsible for internal *dissipation* of energy.

The stability of pHs for the origin equilibrium point $(x^*, u^*) = (0, 0)$ can be determined with the Theorem 2.3.1 using as a Lyapunov candidate function the energy function $H(x)$, then by the properties of $J(x)$ and $R(x)$, the time derivative of $H(x)$ follows that

$$\begin{aligned} \dot{H}(x) =& \nabla^T H(x)\dot{x} \\ =& -\nabla^T H(x)R(x)\nabla H(x) + y^T u \leq u^T y \end{aligned} \tag{2.21}$$

Setting the input $u^* = 0$ converts (2.21) into

$$\dot{H}(x) = -\nabla^T H(x)R(x)\nabla H(x) \leq 0 \tag{2.22}$$

matrix $R(x)$ is symmetric, then (2.22) is always less or equal to zero. Finally, by definition,

pHs are stable in origin.

A last important property of the pHs is the passivity property [6], integrating (2.21) over a time period give us the energy balance equation

$$H(x(t)) - H(x(0)) = \int_0^t \left(-\nabla^T H(x) R(x) \nabla H(x) + y^T u\right) ds \leq \int_0^t u^T y ds \tag{2.23}$$

This last equation indicate us that the energy of pHs is always less or equal than the externally supplied energy. In the case that there are no externally supplied energy the system energy decreases over time if the system presents dissipation matrix

$$H(x(t)) = -\int_0^t \nabla^T H(x) R(x) \nabla H(x) ds + H(x(0)). \tag{2.24}$$

If $H(x)$ is bounded from below, the energy energy system will decrease until reach a minimum energy point.

Example 2.4.1 (**RLC circuit pHs formulation**). *From the physics we know that an RLC circuit presents two elements that storage energy: the capacitor C that accumulates energy in the form of electric charge Q and the Inductor L that accumulates energy as magnetic field ϕ. The energy function of these elements are*

$$H_L(\phi) = \frac{\phi^2}{2L}, \qquad H_C(Q) = \frac{Q^2}{2C},$$

and the total energy of the system is the sum of the energy of every element

$$H(\phi, Q) = \frac{1}{2}\left(\frac{\phi^2}{L} + \frac{Q^2}{C}\right)$$

The gradient of the energy function is

$$\nabla H(\phi, Q) = \begin{bmatrix} \frac{\phi}{L} \\ \frac{Q}{C} \end{bmatrix}.$$

To obtain a pHs formulation for the RLC circuit, it is necessary to rewrite (2.11) *as* (2.20)*, we can do it applying the voltage and current equations*

$$v = \frac{Q}{C}, \qquad i = \frac{\phi}{L}$$

in (2.11)*, then reorganizing give us the pHs state equation*

$$\begin{bmatrix}\dot{\phi}\\ \dot{Q}\end{bmatrix} = \begin{bmatrix}-R & -1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\frac{\phi}{L}\\ \frac{Q}{C}\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix} v_{in} \tag{2.25}$$

we can see that the structural matrices are

$$J(x) = \begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}, \qquad R(x) = \begin{bmatrix}-R & 0\\ 0 & 0\end{bmatrix}, \qquad G(x) = \begin{bmatrix}1\\ 0\end{bmatrix}$$

satisfying the structural conditions. The output equation is calculated by (2.20)*:*

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\frac{\phi}{L}\\ \frac{Q}{C}\end{bmatrix} \tag{2.26}$$

Then the pHs representation is presented as the concatenation of (2.25) *and* (2.26)

$$\Sigma_G : \begin{cases} \begin{bmatrix}\dot{\phi}\\ \dot{Q}\end{bmatrix} = \begin{bmatrix}-R & -1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\frac{\phi}{L}\\ \frac{Q}{C}\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix} v_{in} \\ \\ y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\frac{\phi}{L}\\ \frac{Q}{C}\end{bmatrix} \end{cases} \tag{2.27}$$

The energy ratio is computed as:

$$\begin{aligned}\dot{H}(\phi, Q, v_{in}) &= \begin{bmatrix}\frac{\phi}{L} & \frac{Q}{C}\end{bmatrix}\left(\begin{bmatrix}-R & -1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\frac{\phi}{L}\\ \frac{Q}{C}\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix} v_{in}\right)\\ &= -\frac{R}{L^2}\phi^2 + \frac{\phi}{L} v_{in}\end{aligned}$$

Then, setting $v_{in} = 0$*, we can see that the energy rate is always negative* $\forall(\phi, Q) \neq 0$*, implying that the system is stable at the origin.*

Example 2.4.2 (**Pendulum pHs Formulation**)**.** *Consider the pendulum* (2.5)*. The associated energy function is*

$$H(\theta, \omega) = \frac{1}{2} J\omega^2 + mgl\Big(1 - \cos(\theta)\Big),$$

with gradient

$$\nabla H(\theta, \omega) = \begin{bmatrix} mgl \cdot \sin(\theta) \\ J\omega \end{bmatrix}.$$

The system can be expressed in pHs form as

$$\Sigma_P : \begin{cases} \begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{J} \\ -\frac{1}{J} & -\frac{c}{J^2} \end{bmatrix} \begin{bmatrix} mgl \sin\theta \\ J\omega \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} \tau_{ext}, \\ \quad y = \begin{bmatrix} 0 & \frac{1}{J} \end{bmatrix} \begin{bmatrix} mgl \sin\theta \\ J\omega \end{bmatrix}, \end{cases}$$

where the system matrices are defined as

$$J(x) = \begin{bmatrix} 0 & \frac{1}{J} \\ -\frac{1}{J} & -\frac{1}{J^2} \end{bmatrix}, \quad R(x) = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{c}{J^2} \end{bmatrix}, \quad G(x) = \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix}.$$

The energy ratio is

$$\dot{H}(\theta, \omega, \tau_{ext}) = -c\omega^2 + \omega\tau_{ext}.$$

Finally, setting $\tau_{ext} = 0$, we can see that the energy rate is alwayys negative $\forall(\theta, \omega) \neq 0$, implying that the system is stable at the origin.

2.5 Control Methods

A controller is a dynamical system described by:

$$\begin{aligned} \dot{\xi} &= f_c(\xi, x, r) \\ u &= g_c(\xi, x, r) \end{aligned} \tag{2.28}$$

which, when interconnected with the system (2.3), now called plant, allows for the manipulation of its behaviour so that the plant's output matches a reference signal r over a certain range of values, a diagram of the interconnection is shown in Figure 2.1. In this work, two types of controllers are presented.

2.5.1 Proportional-Integral-Derivative Controllers

Proportional-Integral-Derivative (PID) are linear controllers that feedback the error $r - y$ signal, its derivative and its integral to generate a control action, they are normally represented as an integral differential equation:

$$u = K_P(r - y) + K_I \int (r - y)dt + K_d \frac{d(r - y)}{dt} \tag{2.29}$$

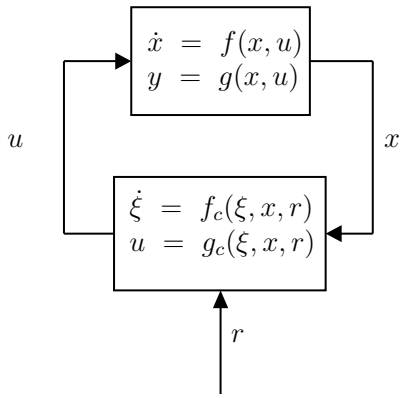


Figure 2.1: General Control Interconnection Diagram

where K_P, K_I and K_D are the controller gains used to design the desired system response. To tune the controller gains multiple techniques can be used depending on the case, the most common ones are the Ziegler-Nichols Oscillation Method and Reaction Curve Based Methods [4], that uses empirical information from the plant to tune the parameters. In this case a different approach is proposed using linear approximations of the system and pole placement methodology.

Pole Placement

Pole Placement is a frequency domain control technique where we use the controller gains to change the position of the plant poles. Let be the plant TF (2.14) and following **Lemma 7.1** in [4], let be A_{cl} an arbitrary polynomial of degree $n_{A_{cl}} = 2n - 1$. Then there exist polynomials $P(s)$ and $L(s)$, with degrees $n_P = n_L = n - 1$ such that

$$E(s)L(s) + F(s)P(s) = A_{cl}(s) \tag{2.30}$$

Then a controller transfer function is obtained by

$$C(s) = \frac{P(s)}{L(s)} \tag{2.31}$$

If r poles and/or zeros wants to be forced on the controller transfer function, then the degrees of the respectively polynomial change into $n_{A_{cl}} = 2n - 1 + r$ and $n_P = n_L = n - 1 + r$.

For the PID, this controller presents the following transfer function

$$PID(s) = K_P + \frac{K_I}{s} + K_D s,$$

not implementable in reality, the derivative term must be approximated by a high-pass filter,

the approximated $PID(s)$ controller

$$PID(s) = K_P + \frac{K_I}{s} + K_D \frac{s}{s+\alpha}, \tag{2.32}$$

is equivalent to a second order bi-proper system. An useful variation of PID is $Proportional-Integral$ (PI) controller, in transfer function form:

$$PI(s) = K_P + \frac{K_I}{s}, \tag{2.33}$$

An implementation scheme is attached

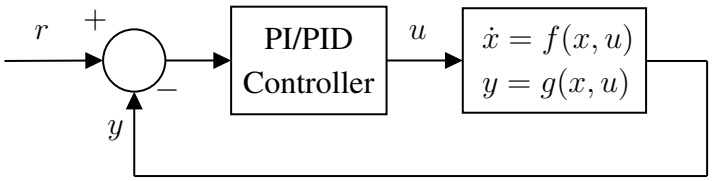


Figure 2.2: PI/PID Control Scheme

Example 2.5.1. ***PID Controller Design:*** *Let's consider the RLC TF* (2.15) *and the* $PID(s)$ *controller* (2.32)*, the* $A_{cl}(s)$ *is given by*

$$A_{cl}(s) = s^4 + \left(\frac{R}{L} + l_0\right)s^3 + \left(\frac{1}{LC} + \frac{Rl_0}{L} + \frac{p_2}{LC}\right)s^2 + \frac{p_1}{LC}s + \frac{p_0}{LC} \tag{2.34}$$

where

$$\begin{aligned} p_2 &= K_P + K_D \\ p_1 &= \alpha K_P + K_I \\ p_0 &= \alpha K_I \\ l_0 &= \alpha \end{aligned}$$

It can be seen that there are four variables to adjust four parameters from the A_{cl}*, then there is a unique solution to the equation system and the poles can be placed arbitrarily. We want to place the closed-loop poles on*

$$A_{cl}(s) = (s+\beta)^4 = s^4 + 4\beta s^3 + 6\beta^2 s^2 + 4\beta^3 s + \beta^4, \tag{2.35}$$

where β *is a desired value. Then, to place the poles at the desired position the following*

equalities must be holds

$$\begin{aligned} \frac{R}{L} + l_0 &= 4\beta \\ \frac{1}{LC} + \frac{Rl_0}{L} + \frac{p_2}{LC} &= 6\beta^2 \\ \frac{p_1}{LC} &= 4\beta^3 \\ \frac{p_0}{LC} &= \beta^4 \end{aligned}$$

Example 2.5.2. ***PI Controller Design:*** *Let's consider the RLC transfer function* (2.15) *and the PI(s) controller* (2.33)*, the* $A_{cl}(s)$ *is given by*

$$A_{cl}(s) = s^3 + \frac{R}{L}s^2 + \left(\frac{1}{LC} + \frac{K_P}{LC}\right)s + \frac{K_I}{LC} \tag{2.36}$$

In this case, there are only two variables available to adjust three parameters, meaning that it is not possible to freely assign all the closed-loop pole locations. However, it is still possible to assign some of the closed-loop poles. The following closed-loop characteristic polynomial is considered:

$$A_{cl}(s) = (s+\beta)(s+\gamma)^2 = s^3 + (2\gamma+\beta)s^2 + \left(\gamma^2 + 2\gamma\beta\right)s + \gamma^2\beta$$

It is observed that β *and* γ *must satisfy:*

$$2\gamma + \beta = \frac{R}{L}$$

Therefore, only one of the two parameters, either β *or* γ*, can be freely chosen, while the other becomes a direct consequence of* (2.5.2)*. Finally, to place the poles the following equalities must be holds*

$$\begin{aligned} \frac{1}{LC} + \frac{K_P}{LC} &= \gamma^2 + 2\gamma\beta \\ \frac{K_I}{LC} &= \gamma^2\beta \end{aligned}$$

2.5.2 Energy Shaping plus Damping Injection Controller

Energy shaping and damping injectio are Passivity Based Control (PBC) [6, 7, 18] that uses the energy properties of pHs to design control systems.

Energy Shaping

Energy Shaping (ES) modifies the energy function $H(x)$ of the original system into a desired energy function $H_d(x)$, whose minimum energy point occurs at $x \neq 0$, thereby shifting the system's equilibrium point to a desired value. This technique seeks to shape the system (2.20) to behave according to target input-state-output pHs:

$$\Sigma_{ES} : \begin{cases} \dot{x} = [J(x) - R(x)]\nabla H_d(x) + G(x)v \\ y = G^T(x)\nabla H_d(x) \end{cases} \tag{2.37}$$

Here, v is an additional input. The desired energy function $H_d(x)$ is assumed to be the sum of the original system's energy and an auxiliary energy function introduced by the controller:

$$H_d(x) = H(x) + H_a(x) \tag{2.38}$$

Substituting (2.38) into (2.20) and equating it to (2.37), we obtain:

$$[J(x) - R(x)]\nabla H_d(x) + G(x)v = [J(x) - R(x)]\nabla H(x) + G(x)u \tag{2.39}$$

Replacing (2.38) into (2.39) and simplifying yields:

$$[J(x) - R(x)]\nabla H_a(x) + G(x)v = G(x)u \tag{2.40}$$

We assume the control input is the sum of a state feedback term and an arbitrary input v:

$$u = \beta(x) + v,$$

Substituting this into (2.40), we get:

$$[J(x) - R(x)]\nabla H_a(x) = G(x)\beta(x). \tag{2.41}$$

The problem now reduces to finding functions $\nabla H_a(x)$ and $\beta(x)$ that satisfy this equation. To solve it, we use **Proposition 7.2.7** from [28]: Applying the full-rank left annihilator $G^{\perp}(x)$ to (2.41) leads to the energy shaping matching equation:

$$G^{\perp}(x)[J(x) - R(x)]\nabla H_a(x) = 0 \tag{2.42}$$

Any $\nabla H_a(x)$ satisfying this equation is a solution of (2.41). The corresponding control input is given by:

$$\beta(x) = (G^T(x)G(x))^{-1}G^T(x)[J(x) - R(x)]\nabla H_a(x). \tag{2.43}$$

Damping Injection

The Damping Injection (DI) uses the input v to extract, or add, power from the system, feed-backing the passive output:

$$v = -Dy + w = -DG^T(x)\nabla H_d(x) + w \tag{2.44}$$

where w is an additional passive input, and $D = D^T > 0$ is the damping injection gain. The DI control action let us increase or decrease the energy rate of (2.37) given by

$$\dot{H}_d(x) = -\nabla^T H_d(x) R(x) \nabla H_d(x) + y^T v \tag{2.45}$$

Replacing (2.37) and (2.44) in (2.45) generates

$$\begin{aligned} \dot{H}_d(x) &= -\nabla^T H_d(x) R(x) \nabla H_d(x) - \left(\nabla^T H_d(x) G(x) D G^T \nabla H_d(x)\right) + y^T w \leq w^T y \\ &= -\nabla^T H_d(x) \left(R(x) + G(x) D G^T(x)\right) \nabla H_d(x) + y^T w \leq w^T y \end{aligned} \tag{2.46}$$

Is important to notice that to maintain the system stability it's necessary that

$$R(x) + G(x) D G^T(x) > 0$$

The DI action then modifies the structure of the original pHs by adding an extra damping matrix through the feedback. The controlled system became:

$$\Sigma_{ESDI} : \begin{cases} \dot{x} = [J(x) - (R(x) + G(x) D G^T(x))] \nabla H_d(x) + G(x) w \\ y = G^T(x) \nabla H_d(x) \end{cases} \tag{2.47}$$

The controller described can be implemented using the control scheme shown in Figure 2.3.

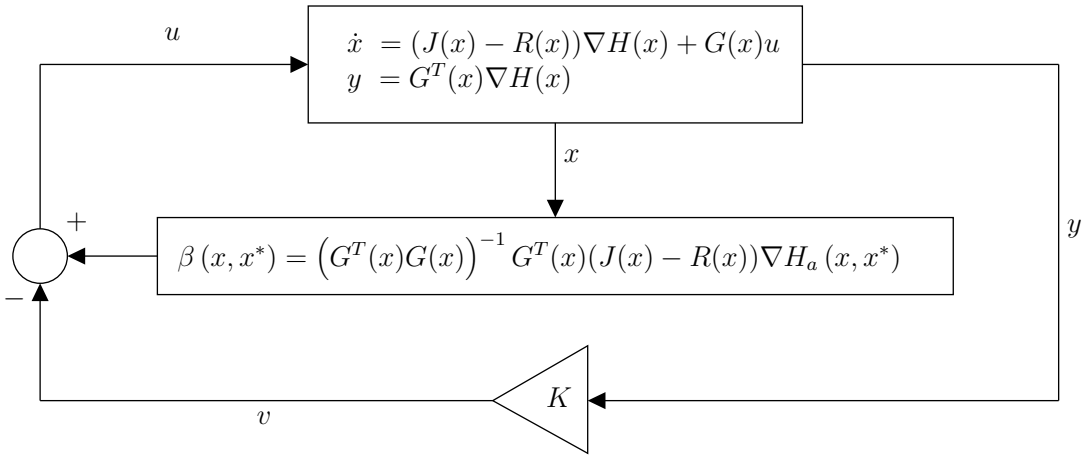


Figure 2.3: Damping Injection + Energy Shaping Control Scheme

Example 2.5.3 (**ESDI Controller Design**). *Consider the pHs RLC circuit* (2.27) *with energy function* (2.4.1)*. The goal is to shift the minimum energy point to a desired equilibrium point:*

$$\phi^* = 0$$
$$Q^* = Cv_{in}^*$$

Applying the ES technique with $G^\perp = \begin{bmatrix} 0 & 1 \end{bmatrix}$ *yields:*

$$\nabla_\phi H_a = 0$$

Hence, the energy function only depends on Q*. The desired energy function is defined as:*

$$\begin{aligned} H_d(\phi, Q) &= \frac{1}{2}\left(\frac{\phi^2}{L} + \frac{(Q - Q^*)^2}{C}\right) \\ &= \frac{1}{2}\left(\frac{\phi^2}{L} + \frac{Q^2}{C}\right) + \frac{1}{2}\left(\frac{(Q^*)^2 - 2QQ^*}{C}\right) \end{aligned}$$

From (2.38)*, we identify:*

$$H_a(Q) = \frac{1}{2}\left(\frac{(Q^*)^2 - 2QQ^*}{C}\right)$$

and its gradient is:

$$\nabla H_a(Q) = -\frac{Q^*}{C}$$

Using this in the control law from (2.43)*, we get:*

$$\beta(Q) = \frac{Q^*}{C}$$

The DI control is applied via output feedback, and the modified damping matrix becomes:

$$R' = \begin{bmatrix} R + D & 0 \\ 0 & 0 \end{bmatrix}$$

The system input thus becomes:

$$v_{in} = \frac{Q^*}{C} - D\frac{\phi}{L} + w$$

The resulting closed-loop system is described by:

$$\Sigma_G : \begin{cases} \begin{bmatrix} \dot{\phi} \\ \dot{Q} \end{bmatrix} = \begin{bmatrix} -(R+D) & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dfrac{\phi}{L} \\ Q - Q^* \\ \hline C \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} w \\ \\ y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \dfrac{\phi}{L} \\ Q - Q^* \\ \hline C \end{bmatrix} \end{cases}$$

$$H_d(\phi, Q) = \frac{1}{2}\left(\frac{\phi^2}{L} + \frac{(Q-Q^*)^2}{C}\right)$$

Chapter 3

Discrete-time Framework

The Continuous-time dynamic models are practically impossible to be implemented in digital systems, as these systems operate by taking data at finite time intervals. For this reason, all digital processing systems in practice use numerical approximations to calculate and simulate the desired dynamical systems.

In this chapter, the different techniques of temporal discretization of dynamic systems and how they can be applied to pHs will be presented. The pros and cons of each type of discretization will be shown, and their performance in approximating continuous-time systems based on accumulated approximation errors will be analysed. In particular, the techniques of Explicit-Euler, formulated by Euler in 1768; Discrete-time Gradient and Collocation Methods as a generalization of the previous methods will be analysed.

3.1 Discrete-Time Systems

A discrete-time dynamical system is described by the general recursive equation

$$x_{k+1} = f_d(k, x_k, u_k)$$

where k denotes the current time step, and x_k and u_k represent the system states and inputs at time step k, respectively. The function $f_d(\cdot)$ defines the system's dynamics. Discrete-time systems also include an output equation, given by:

$$y_k = g_d(k, x_k, u_k)$$

As with continuous-time systems, discrete-time systems can also be described by a state-space model by combining the state and output equations as:

$$\begin{aligned} x_{k+1} &= f_d(k, x_k, u_k) \\ y_k &= g_d(k, x_k, u_k) \end{aligned}$$

A good part of our analysis will deal with state equations without explicit time dependence:

$$\begin{aligned} x_{k+1} &= f_d(x_k, u_k), \\ y_k &= g_d(x_k, u_k), \end{aligned} \tag{3.1}$$

In this case, the system is said to be time-invariant. A state $x = x^*$ is called an equilibrium point if it satisfies $x^* = f_d(x^*, u^*)$ for any arbitrary input u^*.

As in the continuous-time case, an important subclass is given by discrete-time linear systems:

$$\begin{aligned} x_{k+1} &= A_d x_k + B_d u_k \\ y_k &= C_d x_k + D_d u_k \end{aligned} \tag{3.2}$$

where the matrices A_d to D_d are the discrete-time counterparts of matrices A to D in (2.7), preserving the same properties. The linearization process for general systems is carried out in the same way as in the continuous-time case.

3.2 Stability Analysis

The stability concept doesn't change for discrete-time system, but in this time framework, different Lyapunov stability theorems are presented for proving stability [2]. First, let consider the discrete-time autonomous system:

$$x_{k+1} = f_d(x_k) \tag{3.3}$$

where $f_d : \mathbf{D} \to \mathbf{R}^n$ is a locally Lipschitz map from domain $\mathbf{D} \subset \mathbf{R}^n$ into $\mathbf{R}^n$. Suppose $x^* \in \mathbf{D}$ is an equilibrium point of (3.3). The goal is characterize and study the stability of x^*, we will assume that $f_d(0) = 0$ and study the stability the stability of the origin $x = 0$. The following Theorems are presented

Theorem 3.2.1. . *Let $x^* = 0$ be an equilibrium point for* (3.3) *and $\mathbf{D} \subset \mathbf{R}^n$ be a domain*

containing $x^* = 0$. *Let* $V(x_k) : \mathbf{D} \to \mathbf{R}$ *be a continuously differentiable function such that*

$$\begin{aligned} V(0) &= 0, \\ V(x_k) &> 0, \quad \textit{in } \mathbf{D} - \{0\} \\ V(f_d(x_k)) - V(x_k) &\leq 0, \quad \textit{in } \mathbf{D} \end{aligned}$$

Then, $x^* = 0$ *is stable. Moreover, if*

$$V(f_d(x_k)) - V(x_k) < 0, \quad \textit{in } \mathbf{D}, \tag{3.4}$$

then $x^* = 0$ *is asymptotically stable.*

A second stability theorem is derived using the linearization technique.

Theorem 3.2.2. *Let* $x^* = 0$ *be an equilibrium point for the nonlinear system*

$$x_{k+1} = f_d(x_k) \tag{3.5}$$

where $f : \mathbf{D} \to \mathbf{R}^n$ *is continuously differentiable and* $\mathbf{D}$ *is a neighbourhood of the origin. Let*

$$A_d = \left.\frac{df(x)}{dx}\right|_{x=0}$$

Then:

1. *The origin is asymptotically stable if* $|\lambda_i| < 1$ *for all eigenvalues of* A_d.
2. *The origin is unstable if* $|\lambda_i| \geq 1$ *for one or more eigenvalues of* A_d.

3.3 Time-Discretization Methods

Time-discretization consists in obtaining a discrete-time approximation (DTA) of (2.3), i.e., generating a recursive equation (3.3) that presents the same dynamics as the continuous-time one. In general, recursive equations describe the relationship between samples; this relationship is associated with time through the sampling time, denoted by h, which has the following property

$$t = kh.$$

We will start summarizing the DTA method for linear systems and then we will study the principal DTA methods for general systems.

3.3.1 Linear Systems Case

For linear systems, DTA approximations are widely studied [3, 4, 11], there are four important DTA approximations that are worth to study. Let be

$$G(z) = C_d(zI - A_d)^{-1}B_d + D_d \tag{3.6}$$

the discrete time transfer function of (3.2) that results of applying the Zeta transform [3]. Where z is the discrete-time complex frequency variable and I the identity matrix. In this domains the DTA of continuous-time transfer function are performed replacing s by a function of z, depending on the function used different DTA are performed

$$\begin{aligned} \text{Forward Difference:} \quad & s = \frac{z-1}{h} \\ \text{Backward Difference:} \quad & s = \frac{z-1}{zh} \\ \text{Tustin's Approximation:} \quad & s = \frac{2}{h}\frac{z-1}{z+1} \end{aligned} \tag{3.7}$$

Every DTA has their own characteristic that will not be discussed in this work. At least, we have the zero order holder (ZOH) approximation or exact method, it takes in count that between sampling times the signals are constant, mathematically the ZOH can be done by

$$G(z) = \left(1 - z^{-1}\right) \mathcal{Z}\left\{\mathcal{L}^{-1}\left[G(s)\frac{1}{s}\right]\Big|_{t=k\cdot h}\right\} \tag{3.8}$$

where $\mathcal{L}^{-1}$, $\mathcal{Z}$ are the inverse Laplace Transform and the Zeta Transform. This method can be done in time domain for state space systems where we transform the systems matrices

$$A_d = e^{A\cdot h}, \qquad B_d = \int_0^h e^{A\gamma} d\gamma B. \tag{3.9}$$

For C_d and D_d there is no change.

3.3.2 Nonlinear Systems Case

In the case of nonlinear systems, there is no formal procedure to obtain a DTA as in the linear systems. Therefore, the most commonly used approximation methods for ODEs are presented below.

The DTA approximations for ODEs are generally developed for autonomous un-actuated systems (2.16). The extension to non-autonomous and actuated systems will be discussed at the end. All the methods shown below approximate the derivative of the states by a finite

difference over the sampling time

$$\dot{x} \approx \frac{x_{k+1} - x_k}{h}, \tag{3.10}$$

where the main difference among them lies in how they treat the argument of the function $f(\cdot)$. The methods in the following sections are one-step methods, meaning they use data from the current time step to compute the future value [10].

Explicit Euler Method

The Explicit Euler Method is the simplest numerical method. Using (3.10), the function $f(\cdot)$ is evaluated at x_k, yielding

$$x_{k+1} = x_k + hf(x_k).$$

Implicit Euler Method

In the Implicit Euler Method, the system function is evaluated at x_{k+1}, generating an implicit equation

$$x_{k+1} = x_k + hf(x_{k+1}).$$

Since the update equation is implicit, numerical solvers are generally required to compute the solution. An exception arises when applied to a linear system (2.7), for which a closed-form expression can be obtained:

$$x_{k+1} = (I - hA)^{-1} x_k.$$

It should be emphasized that the matrix $I - hA$ must be nonsingular.

Implicit Midpoint Rule

The Implicit Midpoint Rule uses the average value of x_k and x_{k+1} in the argument of the function:

$$x_{k+1} = x_k + hf\left(\frac{x_k + x_{k+1}}{2}\right).$$

As an implicit method, it also requires numerical solvers. However, in the case of linear systems (2.9), an explicit expression can be derived:

$$x_{k+1} = \left(I - \frac{h}{2}A\right)^{-1} \left(I + \frac{h}{2}A\right) x_k.$$

Here again, the matrix $I - \frac{h}{2}A$ must be nonsingular.

Runge-Kutta Methods

The discretization methods study until now were explained exclusively for autonomous un-actuated systems. Runge-Kutta (RK) methods are a generalization of one step methods applicable for non autonomous and actuated systems. Given (2.1), an *s-stage* RK method is computed by

$$\begin{aligned} k_i &= f\left(t_0 + c_i h,\ x_0 + h\sum_{j=1}^{s} a_{ij}k_j,\ u_i\right), \qquad i = 1, .., s \\ x_1 &= x_0 + h\sum_{i=1}^{s} b_i k_i \end{aligned} \tag{3.11}$$

with b_i, a_{ij} and $c_i = \sum_{j=1}^{s} a_{ij}$ real numbers, that are usually displayed as

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s. \end{array}$$

All DTA methods shown previously can be expressed as collocation methods with parameters given by

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \qquad \begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \qquad \begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$$

for the explicit, implicit Euler and midpoint rule respectively. In the analysis of the Runge-Kutta methods the method order is an important property.

Definition 3.3.1. *A Runge-Kutta method has order p, if for all sufficiently regular problems (2.1) the local error $x_1 - x(t_0 + h)$ satisfies*

$$x_1 - x(t_0 + h) = \mathrm{O}(h^{p+1}) \quad as \quad h \to 0$$

Remark 2. *Definition 3.3.1 relates changes in the sampling time τ to changes with the maximum local error. When the sampling time is reduced by a factor of n, the maximum local error decreases proportionally to n^p.*

3.3.3 Collocation Methods

An important family of methods are the *Collocation Methods* (CM), where a polynomial $p(t)$ of degree s is used to approximate the trajectories of (2.1). The polynomial must satisfy:

$$\begin{aligned} p(t_0) &= x_0, \\ \dot{p}(t_0 + c_i h) &= f(t_0 + c_i h,\ p(t_0 + c_i h),\ u(t_0 + c_i h)), \quad i = 1, \dots, s, \end{aligned} \tag{3.12}$$

where c_i are distinct real numbers (usually $0 \leq c_i \leq 1$), called *collocation points*. The numerical solution is then obtained by evaluating the polynomial at the next time step [10]:

$$x_{k+1} = p(t_k + h). \tag{3.13}$$

Example 3.3.1 (**CM polynomial derivation for** $s = 1$)**.** *Let $p(t)$ be a polynomial of degree $s = 1$:*

$$p(t) = a_1 t + a_0, \tag{3.14}$$

and consider a generic ODE

$$\dot{x} = f(t, x, u),$$

with current value $x(t_k) = x_k$. Applying the CM conditions (3.12) *on* (3.14)*, with $x_0 = x_k$ creates the following conditions:*

- ***First condition:*** *Evaluating* (3.14) *at t_k:*

  $$p(t_k) = a_1 t_k + a_0 = x_k,$$

  we compute a_0 as a function of a_1. Replacing in (3.14) *yields:*

  $$p(t) = x_k + (t - t_k)a_1. \tag{3.15}$$

- ***Second condition:*** *Differentiating* (3.15) *gives:*

  $$\dot{p}(t) = a_1,$$

  and applying the second condition yields:

  $$a_1 = f(t_k + c_1 h,\ x_k + c_1 h a_1,\ u(t_k + c_1 h)).$$

Thus, the polynomial for $s = 1$ at each time step k is:

$$\begin{aligned} a_1 &= f(t_k + c_1 h,\ x_k + c_1 h a_1,\ u(t_k + c_1 h)), \\ p(t) &= x_k + (t - t_k)a_1, \end{aligned}$$

and the solution for the next step is:

$$x_{k+1} \approx p(t_k + h) = x_k + h a_1.$$

The Explicit Euler, Implicit Euler and Midpoint Rule are special cases of collocation methods with $c_1 = 0$, $c_1 = 1$ and $c_1 = \frac{1}{2}$, respectively. An important relationship between CM and Runge–Kutta methods is given by:

Theorem 3.3.2 (**Guillou & Soulè (1969), Wright (1970)**). *The collocation method is equivalent to the s-stage Runge–Kutta method* (3.11) *with coefficients*

$$a_{ij} = \int_0^{c_i} l_j(\tau)\, d\tau, \quad b_i = \int_0^1 l_i(\tau)\, d\tau,$$

where $l_i(\tau)$ *is the Lagrange polynomial*

$$l_i(\tau) = \prod_{j \neq i} \frac{(\tau - c_j)}{c_i - c_j}.$$

This method provides an approximation of order s to the exact solution [16]. Common choices for the collocation points include:

1. **Equidistant collocation points:** The collocation points are chosen equidistantly. With s collocation points, the interval is divided into $s+1$ equal parts. Examples for $s = 2$ and $s = 3$ are:

$\frac{1}{3}$	$\frac{1}{2}$	$-\frac{1}{6}$
$\frac{2}{3}$	$\frac{2}{3}$	0
	$\frac{1}{2}$	$\frac{1}{2}$

Table 3.1: $s = 2$ equidistant collocation points parameters.

$\frac{1}{4}$	$\frac{23}{48}$	$-\frac{1}{3}$	$\frac{5}{48}$
$\frac{2}{4}$	$\frac{7}{12}$	$-\frac{1}{6}$	$\frac{1}{12}$
$\frac{3}{4}$	$\frac{9}{16}$	0	$\frac{3}{16}$
	$\frac{2}{3}$	$-\frac{1}{3}$	$\frac{2}{3}$

Table 3.2: $s = 3$ equidistant collocation points parameters.

2. **Gaussian quadrature points:** The collocation points are chosen as the zeros of the shifted Legendre polynomial of degree s:

$$\frac{d^s}{dx^s}\left(x^s(x-1)^s\right).$$

The resulting method has order $2s$. Parameters for $s = 2$ and $s = 3$ are shown below:

$\frac{1}{2} - \frac{\sqrt{3}}{6}$	$\frac{1}{4}$	$\frac{1}{4} - \frac{\sqrt{3}}{6}$
$\frac{1}{2} + \frac{\sqrt{3}}{6}$	$\frac{1}{4} + \frac{\sqrt{3}}{6}$	$\frac{1}{4}$
	$\frac{1}{2}$	$\frac{1}{2}$

Table 3.3: $s = 2$ Gauss collocation points parameters.

$\frac{1}{2} - \frac{\sqrt{15}}{10}$	$\frac{5}{36}$	$\frac{2}{9} - \frac{\sqrt{15}}{15}$	$\frac{5}{36} - \frac{\sqrt{15}}{30}$
$\frac{1}{2}$	$\frac{5}{36} + \frac{\sqrt{15}}{24}$	$\frac{2}{9}$	$\frac{5}{36} - \frac{\sqrt{15}}{24}$
$\frac{1}{2} + \frac{\sqrt{15}}{10}$	$\frac{5}{36} + \frac{\sqrt{15}}{30}$	$\frac{2}{9} + \frac{\sqrt{15}}{15}$	$\frac{5}{36}$
	$\frac{5}{18}$	$\frac{4}{9}$	$\frac{5}{18}$

Table 3.4: $s = 3$ Gauss collocation points parameters.

Finally, the CM method can be summarized by the following system of equations:

$$\begin{aligned} x_{c_i} &= x_k + h\sum_{j=1}^{s} a_{ij} f(t_{c_j}, x_{c_j}, u_{c_j}), \quad i = 1, \dots, s, \\ x_{k+1} &= x_k + h\sum_{j=1}^{s} b_j f(t_{c_j}, x_{c_j}, u_{c_j}). \end{aligned} \tag{3.16}$$

Example 3.3.3. *Let's consider* (2.12) *where*

$$x_{ci} = \begin{bmatrix} i_{ci} \\ v_{ci} \end{bmatrix}, \qquad f(x_{ci}, u_{ci}) = \begin{bmatrix} -\frac{R}{C} i_{ci} - \frac{1}{L} v_{ci} + \frac{1}{L} v_{ci}^{in} \\ \frac{1}{C} i_{ci} \end{bmatrix},$$

and two collocation points $c_1 = \frac{1}{3}$ *and* $c_2 = \frac{2}{3}$ *that generates*

$$l_1(\tau) = \frac{\tau - \frac{2}{3}}{\frac{1}{3} - \frac{2}{3}} = -3\left(\tau - \frac{2}{3}\right) \qquad l_2(\tau) = \frac{\tau - \frac{1}{3}}{\frac{2}{3} - \frac{1}{3}} = 3\left(\tau - \frac{1}{3}\right)$$

$$a_{11} = \int_0^{\frac{1}{3}} -3\left(\tau - \tfrac{2}{3}\right) d\tau = \tfrac{1}{2} \qquad a_{12} = \int_0^{\frac{2}{3}} -3\left(\tau - \tfrac{2}{3}\right) d\tau = -\tfrac{1}{6}$$

$$a_{21} = \int_0^{\frac{2}{3}} 3\left(\tau - \tfrac{1}{3}\right) d\tau = \tfrac{2}{3} \qquad a_{22} = \int_0^{\frac{2}{3}} 3\left(\tau - \tfrac{1}{3}\right) d\tau = 0$$

$$b_1 = \int_0^{1} -3\left(\tau - \tfrac{2}{3}\right) d\tau = \tfrac{1}{2} \qquad b_1 = \int_0^{1} 3\left(\tau - \tfrac{1}{3}\right) d\tau = \tfrac{1}{2}$$

The solution given by CM is calculated by:

$$\begin{aligned} x_{c1} &= x_k + \tau a_{11} f(x_{c1}, u_{c1}) + \tau a_{12} f(x_{c2}, u_{c2}) \\ x_{c2} &= x_k + \tau a_{21} f(x_{c1}, u_{c1}) \\ x_{k+1} &= x_k + \tau b_1 f(x_{c1}, u_{c1}) + \tau b_2 f(x_{c2}, u_{c2}) \end{aligned}$$

3.4 Time-Discretization for port-Hamiltonian Systems

Once the time-discretization methods for dynamical systems have been presented, the next step is to implement them in pHs while preserving their structural properties, such as the matrices $J(x)$ and/or $R(x)$, or energetic properties such as the energy rate $\dot{H}(x)$ or the total energy of the system. Therefore, it is necessary to find an approach that satisfies both objectives.

The pHs present two derivative terms: the time derivative of the states $\dot{x}$ and the gradient of the energy function $\nabla H(x)$. To obtain a discrete-time representation, it is necessary to address the discretization of both terms. In the case of the time derivative, multiple

methodologies were presented in the previous section. In the case of the gradient, according to the literature [15, 17, 19], two approaches can be considered:

1. **Treat the term as an algebraic expression:** This option allows us to regard the term $(J(x)-R(x))\nabla H(x)$ as an arbitrary function $f(x)$, and thus treat the pHs system as
$$\dot{x}(t) = f(x) + g(x)u$$
This approach enables the use of any of the previously presented time-discretization methods.

2. **Derive an equivalent expression of the gradient for discrete time:** This approach seeks to preserve certain properties between the continuous-time and discrete-time representations.

In both cases, it is necessary to analyse which properties are preserved.

3.4.1 Discrete Gradient Method

The discrete Gradient (dG) method uses the definition of discrete derivative, proposed in [29], to approximate the gradient of the energy function $\nabla H(x)$. This derivative may hold the properties of the following definition.

Definition 3.4.1. *A discrete derivative for a smooth function* $f : \mathrm{R}^n \to \mathrm{R}$ *is a mapping* $\bar{\nabla} f : \mathrm{R}^n \times \mathrm{R}^n \to \mathrm{R}^n$ *with the following properties:*

1. *Directionality.* $(y - x)^T \bar{\nabla} f(x, y) = f(y) - f(x)$ *for any* x, $y \in \mathrm{P}$.
2. *Consistency.* $\lim_{y \to x} \bar{\nabla} f(x, y) = \nabla f(x)$ *for all* x, $y \in \mathrm{P}$

Some examples of dG are:

- **Mean Value Discrete Gradient** [30]
$$\bar{\nabla} H(x_k, x_{k+1}) = \int_0^1 \nabla H((1-\sigma)x_k + \sigma x_{k+1}) d\sigma \tag{3.17}$$

- **Gonzales Discrete Gradient** [29]
$$\begin{aligned} \bar{\nabla} H(x_k, x_{k+1}) = {} & \nabla H\left(\frac{x_k + x_{k+1}}{2}\right) \\ & + \frac{H(x_{k+1}) - H(x_k) - \nabla^T H\left(\frac{x_k + x_{k+1}}{2}\right)(x_{k+1} - x_k)}{||(x_{k+1} - x_k)||^2}(x_{k+1} - x_k) \end{aligned} \tag{3.18}$$

A particular case of discrete gradient arises when it takes a quadratic form, such as $H(x) = \frac{1}{2} x^T P x$. According to [31], we have:

Lemma 3.4.1. *Given $H(x) = \frac{1}{2}x^T Px$, with symmetric P matrix, then the associated discrete gradient verifies*

$$\bar{\nabla}H(x_k, x_{k+1}) = \frac{1}{2}P\left(x_k + x_{k+1}\right) = \frac{1}{2}\left(\nabla H(x_k) + \nabla H(x_{k+1})\right) \tag{3.19}$$

Proof. This is directly deduced from the equality

$$\begin{aligned} H(x_{k+1}) - H(x_k) &= \frac{1}{2}x_{k+1}^T Px_{k+1} - \frac{1}{2}x_k^T Px_k \\ &= \frac{1}{2}\left((x_{k+1} - x_k)^T P(x_{k+1} - x_k)\right). \end{aligned}$$

■

With the defined approximations, the dt-pHs is defined following [18]:

Definition 3.4.2. *Let be*

$$\bar{\Sigma}_G : \begin{cases} x_{k+1} = x_k + h[\bar{J}(x_k, x_{k+1}) - \bar{R}(x_k, x_{k+1})]\bar{\nabla}H(x_k, x_{k+1}) + hG(x_k, x_{k+1})u_k \\ y_k = G^T(x_k, x_{k+1})\bar{\nabla}H(\hat{x}) \end{cases} \tag{3.20}$$

the discrete-time approximation of pHs (2.20) *obtained using dG. With sampling time h, energy function $H(x_k)$ and the change of the energy function defined with the discrete Gradient $\bar{\nabla}H(x_k, x_{k+1})$. Where $u_k = u(kh)$, with $k \in \mathrm{N}$ the input at the sampling instant, matrices $[\bar{J}(x_k, x_{k+1}) - \bar{R}(x_k, x_{k+1})] : \mathrm{R}^n \times \mathrm{R}^n \to \mathrm{R}^{n\times n}$ and $G(x_k, x_{k+1}) : \mathrm{R}^n \times \mathrm{R}^n \to \mathrm{R}^{n\times m}$ the discrete-time approximation of $[J(x) - R(x)]$ and $G(x)$, respectively, where $\forall x_k, x_{k+1} \in \mathrm{R}^n$ we have that*

$$\begin{aligned} [\bar{J}(x_k, x_k) - \bar{R}(x_k, x_k)] &= [J(x) - R(x)] \\ \bar{G}(x_k, x_k) &= G(x) \end{aligned} \tag{3.21}$$

$$[\bar{J}(x_k, x_k) - \bar{R}(x_k, x_k)] + [\bar{J}(x_k, x_k) - \bar{R}(x_k, x_k)]^T \leq 0. \tag{3.22}$$

The system matrices are a sort of degree of freedom choice for the designer, some admissible options are:

$$\begin{aligned} [\bar{J}(x_k, x_{k+1}) - \bar{R}(x_k, x_{k+1})] = [J(x_k) - R(x_k)] \quad or \quad \left[J\left(\frac{x_k + x_{k+1}}{2}\right) - R\left(\frac{x_k + x_{k+1}}{2}\right)\right] \\ \bar{G}(x_k, x_{k+1}) = G(x_k) \quad or \quad G\left(\frac{x_k + x_{k+1}}{2}\right) \end{aligned} \tag{3.23}$$

The change of energy in the system is derived from the directionality propertie of the discrete

derivative:

$$\frac{H(x_{k+1}) - H(x_k)}{h} = -\bar{\nabla}^T H(x_k, x_{k+1}) \bar{R}(x_k, x_{k+1}) \bar{\nabla} H(x_k, x_{k+1}) + y_k^T u_k \leq u_k^T y_k \tag{3.24}$$

The stability of the DTA is proved applying Theorem 3.2.1 on (3.24) with $u_k = 0$. We can see that (3.24) is always negative, then the DTA is stable. It is worth noting that if the energy function is quadratic, $H = \frac{1}{2} x^t P x$, and the matrices of the discrete system depend on the midpoint argument, $\frac{x_{k+1}+x_k}{2}$, the discrete Gradient method is equivalent to the Implicit Midpoint Rule.

Remark 3. *The dG DTA preserves the structure of the continuous-time pHs and the stability of the system.*

Example 3.4.2. *A DTA for RLC system described by* (2.27) *using dt-G is presented. This system presents a quadratic energy function, using the Lemma 3.4.1, a dG for the system is*

$$\bar{\nabla} H(\phi, Q) = \begin{bmatrix} \frac{\phi_k + \phi_{k+1}}{2L} \\ \frac{Q_k + Q_{k+1}}{2C} \end{bmatrix}$$

and the dt-G pHs representation for RLC circuit is shown as follows

$$\bar{\Sigma}_G : \begin{cases} \begin{bmatrix} \phi_{k+1} \\ Q_{k+1} \end{bmatrix} = \begin{bmatrix} \phi_k \\ Q_k \end{bmatrix} + \tau \begin{bmatrix} -R & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{\phi_k + \phi_{k+1}}{2L} \\ \frac{Q_k + Q_{k+1}}{2C} \end{bmatrix} + \tau \begin{bmatrix} 1 \\ 0 \end{bmatrix} v_k^{in} \\ \\ y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{\phi_k + \phi_{k+1}}{2L} \\ \frac{Q_k + Q_{k+1}}{2C} \end{bmatrix} \end{cases}$$

The methodology presented can be applied exclusively to pHs. If a system does not exhibit the required structure, it is not possible to obtain a DTA using this approach. This limitation becomes particularly relevant when implementing non-passive controllers, as these typically break the pHs structure. For this reason, the approach presented here is not suitable for the discretization of controllers.

3.4.2 Collocation Methods

As presented in [15, 19]. Using CM, a DTA of pHs is done applying (3.16) in (2.20), resulting in:

$$\begin{aligned} x_{c_i} &= x_k + h \sum_{j=1}^{s} a_{ij} \left(\left[J(x_{c_j}) - R(x_{c_j}) \right] \nabla H(x_{c_j}) + G(x_{c_j}) u_{cj} \right), \qquad i = 1, \ldots, s \\ x_{k+1} &= x_k + h \sum_{j=1}^{s} b_j \left(\left[J(x_{c_j}) - R(x_{c_j}) \right] \nabla H(x_{c_j}) + G(x_{c_j}) u_{cj} \right) \end{aligned} \tag{3.25}$$

The output is computed for every collocation point and at the next time step $k+1$ as:

$$\begin{aligned} y_{ci} &= G^T(x_{c_i})\nabla H(x_{c_i}), \qquad i = 1, \ldots, s \\ y_{k+1} &= G^T(x_{k+1})\nabla H(x_{k+1}) \end{aligned} \tag{3.26}$$

In this discretization method, the pHs is threat as a generic ODE. To obtain a complete discrete-time pHs formulation requires defining the associated energy function and deriving an expression for the energy rate to prove stability. For this porpoise, the following Proposition is presented.

Proposition 1. *The dt-pHs approximation presented in* (3.25) *and* (3.26) *features the same energy function $H(x)$, but evaluated over the trajectories of the dt-pHs. The change of energy over the time is computed approximating* (2.21) *using collocation methods*

$$H(x_{k+1}) = H(x_k) + h\sum_{j=1}^{s} b_j \left(-\nabla H^T(x_{c_j})R(x_{c_j})\nabla H(x_{c_j}) + y_{cj}^T u_{cj}\right) \tag{3.27}$$

Proof. Applying CM (3.16) to (2.21) generates the following equations

$$\begin{aligned} H(x_{c_i}) &= H(x_k) + h\sum_{j=1}^{s} a_{ij}\left[-\nabla H^T(x_{c_j})R(x_{c_j})\nabla H(x_{c_j}) + y_{cj}^T u_{cj}\right], \quad i = 1, \ldots, s \\ H(x_{k+1}) &= H(x_k) + h\sum_{j=1}^{s} b_j\left[-\nabla H^T(x_{c_j})R(x_{c_j})\nabla H(x_{c_j}) + y_{cj}^T u_{cj}\right] \end{aligned}$$

The first equation calculates the energy function at the collocation points meanwhile the second one calculates the energy function at the next time step. ■

Stability analysis can be developed using the discrete-time Lyapunov Theory shown in Theorem 3.2.1. Let's consider the discrete-time pHs shown in (3.25) where $u_{ci} = 0 \; \forall \; ci$ and using as a candidate Lyapunov function $H(x_k)$, the rate of change of Lyapunov function over the time

$$H(x_{k+1}) - H(x_k) = -h\sum_{j=1}^{s} b_j \nabla H^T(x_{c_j})R(x_{c_j})\nabla H(x_{c_j}) \leq 0 \tag{3.28}$$

is always negative because $b_j \geq 0$ and $R(x_{c_j})$ by definition is symmetric, then the term $\nabla H^T(x_{c_j})R(x_{c_j})\nabla H(x_{c_j})$ is always positive, then $x = 0$ is stable.

Remark 4. *The CM DTA of pHs doesn't preserve the structure of the continuous-time pHs, the state equation is replaced by* (3.25)*. The stability property in the origin is translated to the discrete-time domain by* (3.27)*. The principal advantage of this methodology is the higher approximation order that arise from CM.*

Example 3.4.3. *Let's consider the RLC-series circuit from the previous example and two Gaussian Collocation Points from Table 3.3, a discrete-time approximation is performed as*

$$\begin{bmatrix}\phi_{c1}\\ Q_{c1}\end{bmatrix} = \begin{bmatrix}\phi_k\\ Q_k\end{bmatrix} + h\left[\frac{1}{4}\left(\begin{bmatrix}-R & -1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\phi_{c1}\\ Q_{c1}\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix} v_{c1}^{in}\right) + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)\left(\begin{bmatrix}-R & -1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\phi_{c2}\\ Q_{c2}\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix} v_{c2}^{in}\right)\right]$$

$$\begin{bmatrix}\phi_{c2}\\ Q_{c2}\end{bmatrix} = \begin{bmatrix}\phi_k\\ Q_k\end{bmatrix} + h\left[\left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)\left(\begin{bmatrix}-R & -1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\phi_{c1}\\ Q_{c1}\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix} v_{c1}^{in}\right) + \frac{1}{4}\left(\begin{bmatrix}-R & -1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\phi_{c2}\\ Q_{c2}\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix} v_{c2}^{in}\right)\right]$$

used to calculate the states at the collocation points and

$$\begin{bmatrix}\phi_{k+1}\\ Q_{k+1}\end{bmatrix} = \begin{bmatrix}\phi_k\\ Q_k\end{bmatrix} + h\left[\frac{1}{2}\left(\begin{bmatrix}-R & -1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\phi_{c1}\\ Q_{c1}\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix} v_{c1}^{in}\right) + \frac{1}{2}\left(\begin{bmatrix}-R & -1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\phi_{c2}\\ Q_{c2}\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix} v_{c2}^{in}\right)\right]$$

used to computed the state at the next step-time.

3.5 Time-Approximation for Controllers

A discrete-time approximation can be developed using CM (3.16) to (2.28), obtaining

$$\begin{aligned}\xi_{c_i} &= \xi_k + h\sum_{j=1}^{n} a_{ij} f_c(\xi_{c_j}, x_{c_j}, r_{c_j}) \qquad i = 1, ..., n\\ \xi_{k+1} &= \xi_k + h\sum_{i=1}^{n} b_i f_c(\xi_{c_i}, x_{c_i}, r_{c_i})\end{aligned} \quad . \tag{3.29}$$

the control action then is calculated by

$$u_k = g_c(\xi_k, x_k, r_k). \tag{3.30}$$

To apply the presented controller DTA, the state of the plant at the collocation points x_{ci} are needed, this is non-causal, i.e. impossible to apply in the real applications. To overcome the problem we can pre-compute the resulting closed loop system when the input from (2.28) is applied to (2.3) [17–19], we will refer to this approach as *Target System Strategy*.

Target System Strategy

In Target System Strategy (TSS) we calculate the resulting closed-loop system when u of (2.28) is applied to (2.3)

$$\begin{bmatrix}\dot{x}\\ \dot{\xi}\end{bmatrix} = \begin{bmatrix}f(x, g_c(\xi, x, r))\\ f_c(\xi, x, r)\end{bmatrix}$$

then, rewriting the system as

$$\dot{x}^{ext} = f_{ext}(x^{ext}, r). \tag{3.31}$$

a DTA can be developed using CM (3.16) to (3.31) obtaining

$$\begin{aligned} x_{c_i}^{ext} &= x_k^{ext} + h\sum_{j=1}^{n} a_{ij} f_{ext}(x_{c_j}^{ext}, r_{c_j}), \qquad i = 1, ..., n \\ x_{k+1}^{ext} &= x_k^{ext} + h\sum_{i=1}^{n} b_i f_{ext}(x_{c_i}^{ext}, r_{c_i}) \end{aligned} \quad . \tag{3.32}$$

The first equation in (3.32) allows the computation of both the plant and controller states at the collocation points c_i. This requires the value of the reference signal at the collocation points, r_{c_i}, which is not a limitation since this signal is assumed to be known for every k, and the value of the states at the present instant k. Then, using the second equation, all the states for the next time step $k+1$ are computed. Finally, the controller action is computed by (3.30).

The TSS allows the controller dynamics to be implemented implicitly, since at each time step k, the value of ξ_{k+1} is computed. However, it introduces a trade-off that must be considered. While it removes the dependency on a non-causal variable, it requires knowledge of the system states at the current time step. This necessitates the use of a state estimator if the states cannot be directly measured. The corresponding control diagram is provided below:

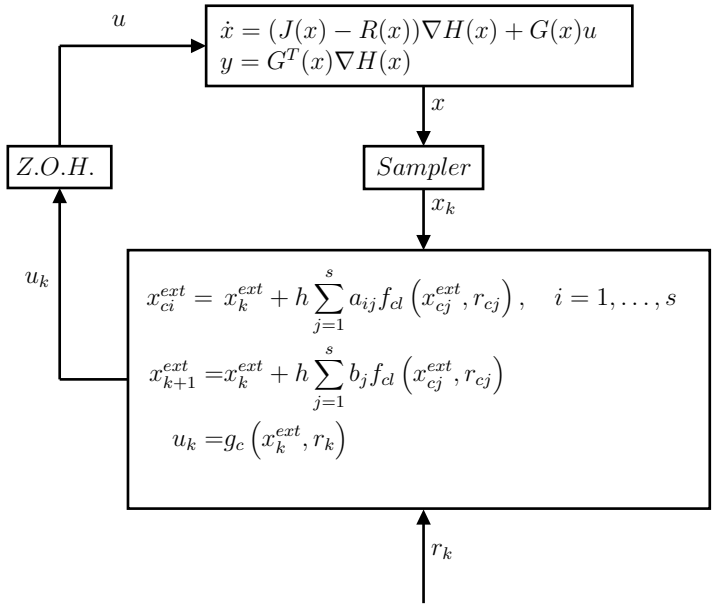


Figure 3.1: Control diagram illustrating the implementation of TSS.

Example 3.5.1. *Consider the RC series circuit described by*

$$\begin{aligned} \dot{v}_c &= -\frac{1}{RC} v_c + \frac{1}{RC} v_f \\ y &= v_c \end{aligned} \tag{3.33}$$

and the PI controller

$$\dot{\xi} = r - y$$
$$u = K_P(r - y) + K_I\xi.$$

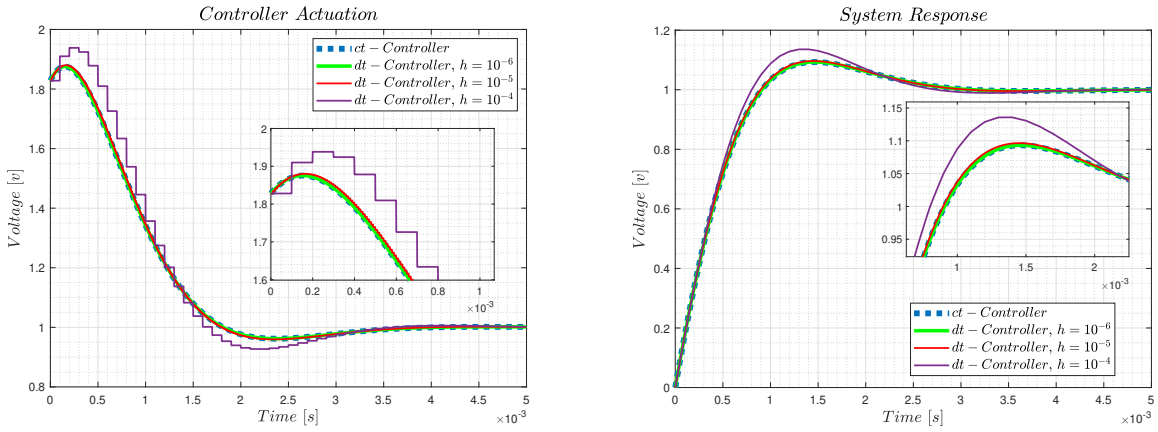
The target system is obtained applying the control input into the system generating

$$\begin{bmatrix} \dot{v}_c(t) \\ \dot{\xi}(t) \end{bmatrix} = \begin{bmatrix} -\frac{1}{RC}(K_P + 1) & \frac{K_I}{RC} \\ -1 & 0 \end{bmatrix} \begin{bmatrix} v_c(t) \\ \xi(t) \end{bmatrix} + \begin{bmatrix} \frac{K_P}{RC} \\ 1 \end{bmatrix} r(t) \tag{3.34}$$

Using as parameters for the plant and controller

$$R = 1 \quad C = 0.001$$
$$K_P = 1.828 \quad K_I = 4000,$$

step response can be simulated for continuos-time and discrete-time controllers, for different values of h:



(a) Controller Actuation **(b)** System Response

Figure 3.2: Example 2.3.1 Time-response

Chapter 4

The Piezoelectric Actuator Case

The piezoelectric phenomenon entails the ability to generate an electric charge as a reaction to mechanical stress or deformation and, in the opposite way, to deform when exposed to an electric field [20]. Piezoelectric Actuators (PZA) are electromechanical systems that exploit the piezoelectric effect to generate micrometric-scale deformations by applying high voltages, enabling precise position control. However, these actuators exhibit hysteresis in their behaviour, a highly nonlinear input output relationship that difficult the system modelling. The modelling of hysteresis phenomena has been widely studied, classical modelling approaches include the Prandtl–Ishlinskii model [32], based on $min(\cdot)$ and $max(\cdot)$ operators, and the Bouc–Wen model [24], which uses input time-derivatives and absolute values of them, both yield non smooth models. Alternatively, hysteresis modelling in pHs framework are developed in [26], which models this behaviour using irreversible pHs, and [27, 33], employing hysterons [21], i.e., energy storage elements coupled with nonlinear dissipation, interconnected with a mass–spring–damper system through Bond–Graph methods [8].

4.1 Hysteron Modelling Approach

Hysterons are systems composed of energy-storing elements interconnected with nonlinear dissipative elements, which allow modelling hysteresis curves [21]. This approach provides a systematic framework for approximating hysteresis loops through modular components. For the present work, we employ an RC parallel circuit hysteron model:

$$\dot{Q} = -h^{-1}\left(\frac{Q}{C}\right) + i_h,$$

where Q represents the electric charge stored in a capacitor with capacitance C, i_h denotes the input current and $h^{-1}(\cdot)$ corresponds to a nonlinear admittance function. Specifically,

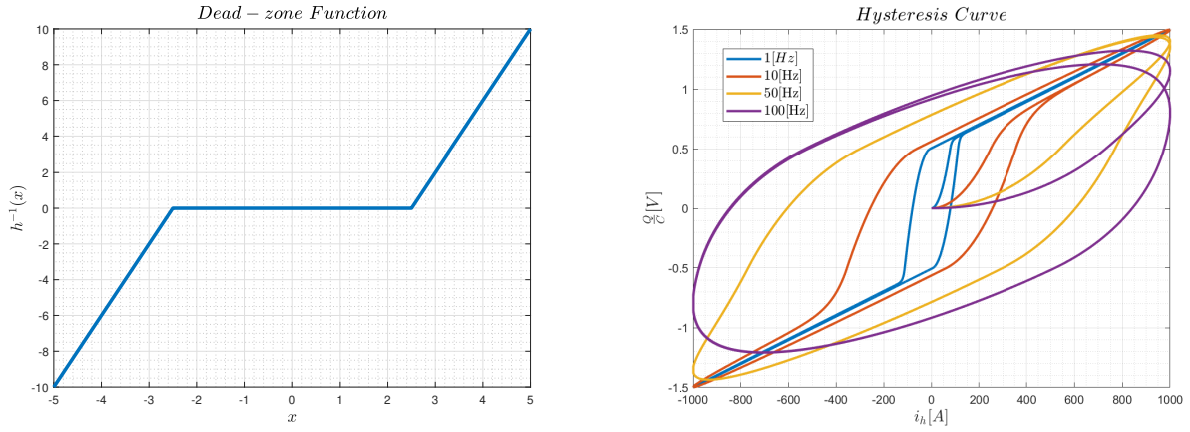
we implement a dead-zone nonlinearity:

$$h^{-1}\left(\frac{Q}{C}\right) = \begin{cases} \rho^{-1}\left(\frac{Q}{C} + \frac{d}{2}\right), & \frac{Q}{C} < -\frac{d}{2} \\ 0, & -\frac{d}{2} \leq \frac{Q}{C} \leq \frac{d}{2} \\ \rho^{-1}\left(\frac{Q}{C} - \frac{d}{2}\right), & \frac{d}{2} < \frac{Q}{C} \end{cases}$$

where ρ^{-1} represents the resistive element's admittance and d defines the deadzone width. Figure 4.1a illustrates this characteristic nonlinearity. The resulting hysteresis behavior, generated with parameter values:

$$\begin{aligned} \rho^{-1} &= 100\ [\Omega^{-1}] \\ d &= 10\ [\mathrm{V}] \\ C &= 0.1\ [\mathrm{F}], \end{aligned}$$

demonstrates frequency-dependent characteristics as shown in Figure 4.1b.



(a) Characteristic deadzone function $h^{-1}(x)$ **(b)** Frequency-dependent hysteresis loops (1-100 Hz)

Figure 4.1: Hysteron nonlinearity characterization

A key advantage of the hysteron paradigm lies in its modular architecture. Multiple hysterons can be interconnected to construct complex hysteresis patterns, as shown in Figure 4.2. This scalability enables precise matching of diverse hysteresis phenomena observed in piezoelectric systems.

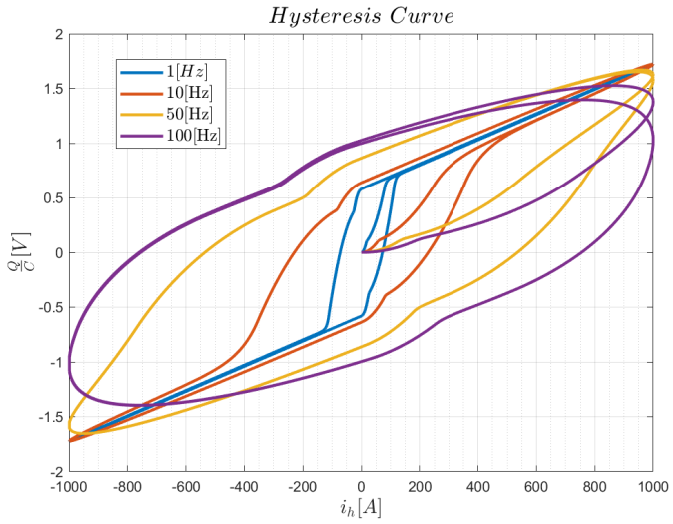


Figure 4.2: Composite hysteresis loop generated by four interconnected hysterons

4.2 Piezoelectric Actuator Model

A modular pHs model of the Piezoelectric Actuator (PZA) was developed in [27,33] using the Bond-Graphs methodology [8], which allowed integrating n hysterons into a mass-spring-damper mechanical model. In this work, a model with two hysterons is used, described by:

$$\Sigma_G : \begin{cases} \begin{bmatrix} \dot{Q}_1 \\ \dot{Q}_2 \\ \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} -h_1^{-1}\left(\frac{Q_1}{C_1}\right)\frac{C_1}{Q_1} & 0 & 0 & \alpha \\ 0 & -h_2^{-1}\left(\frac{Q_2}{C_2}\right)\frac{C_2}{Q_2} & 0 & \alpha \\ 0 & 0 & 0 & 1 \\ -\alpha & -\alpha & -1 & -b \end{bmatrix} \begin{bmatrix} \frac{Q_1}{C_1} \\ \frac{Q_2}{C_2} \\ qk \\ \frac{p}{m} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \alpha \end{bmatrix} v_{in} \\ y = \begin{bmatrix} 0 & 0 & 0 & \alpha \end{bmatrix} \begin{bmatrix} \frac{Q_1}{C_1} \\ \frac{Q_2}{C_2} \\ qk \\ \frac{p}{m} \end{bmatrix} \end{cases}, \tag{4.1}$$

where C_i is the capacitance of the i-th hysteron, k is the elastic stiffness, m is the mass, b the damping, and α is a transformation constant. The system variables are Q_i the charge of the Hysteron i, q the PZA position and p the momenta. The input v_{in} is the voltage applied to the system. The energy function

$$H(x) = \frac{1}{2}\left(\frac{Q_1^2}{C_1} + \frac{Q_2^2}{C_2} + kq^2 + \frac{p^2}{m}\right) \tag{4.2}$$

has a minimum at the origin. The system presents the equilibrium points described by

$$\begin{aligned} h_1^{-1}\left(\frac{Q_1^*}{C_1}\right) &= 0 \\ h_2^{-1}\left(\frac{Q_2^*}{C_2}\right) &= 0 \\ p^* &= 0 \\ q^* &= \frac{\alpha}{k}\left(-\frac{Q_1^*}{C_1} - \frac{Q_2^*}{C_2} + V_{in}\right) \end{aligned} \quad . \tag{4.3}$$

Due to the nonlinear function $h_i^{-1}(\cdot)$, the system presents infinite number of equilibrium points that doesn't match the minimum energy point when the input is zero. In [27], a parameter table was obtained:

Parameter	Value	Units
m	1.0148×10^{-3}	[kg]
k	24579	[N/m]
b	3.7356	[Ns/m]
C_1	5.6425×10^{-5}	$[F]$
ρ_1^{-1}	0.6002	$[\Omega^{-1}]$
d_1	14.7571	$[V]$
C_2	5.2125×10^{-7}	$[F]$
ρ_2^{-1}	1.1528×10^{-4}	$[\Omega^{-1}]$
d_2	8.6838	$[V]$
α	0.046311	$[C/m]$

Table 4.1: Parameters for the pHs model.

These parameters will be used for controller design process and for the development of simulations throughout the remainder of this chapter.

4.3 Discrete-time Model

Two discrete-time models are created using the methodologies shown in Sections 3.4.1 and 3.4.2, then step-responses are performed simulating and error analysis is performed.

Discrete-time Gradient Models

The first discrete-time model to be develop is generated using dG methodology. For this we need to determine the discrete Gradient to be used and the form of the system matrices. Regarding the first point, the Mean Value discrete Gradient (3.17) is selected due to its

computational simplicity:

$$\bar{\nabla}H(x_k, x_{k+1}) = \begin{bmatrix} \frac{1}{C_1}\left(\frac{Q_k^1 + Q_{k+1}^1}{2}\right) \\ \frac{1}{C_2}\left(\frac{Q_k^2 + Q_{k+1}^2}{2}\right) \\ k\left(\frac{q_k + q_{k+1}}{2}\right) \\ \frac{1}{m}\left(\frac{p_k + p_{k+1}}{2}\right) \end{bmatrix} = \frac{1}{2}K(x_k + x_{k+1})$$

where $K = \text{diag}\left(\frac{1}{C_1}, \frac{1}{C_2}, k, \frac{1}{m}\right)$ is a diagonal matrix. For the system matrices, both options presented in (3.23) are considered:

$$\left[\bar{J}(x_k) - \bar{R}(x_k)\right] = \begin{bmatrix} -h_1^{-1}\left(\frac{Q_k^1}{C_1}\right)\frac{C_1}{Q_k^1} & 0 & 0 & \alpha \\ 0 & -h_2^{-1}\left(\frac{Q_k^2}{C_2}\right)\frac{C_2}{Q_k^2} & 0 & \alpha \\ 0 & 0 & 0 & 1 \\ -\alpha & -\alpha & -1 & -b \end{bmatrix}$$

$$\left[\bar{J}(x_k, x_{k+1}) - \bar{R}(x_k, x_{k+1})\right] = \begin{bmatrix} -h_1^{-1}\left(\frac{Q_k^1 + Q_{k+1}^1}{2C_1}\right)\frac{2C_1}{Q_k^1 + Q_{k+1}^1} & 0 & 0 & \alpha \\ 0 & -h_2^{-1}\left(\frac{Q_k^2 + Q_{k+1}^2}{2C_2}\right)\frac{2C_2}{Q_k^2 + Q_{k+1}^2} & 0 & \alpha \\ 0 & 0 & 0 & 1 \\ -\alpha & -\alpha & -1 & -b \end{bmatrix}$$

Collocation Method Models

The second discrete-time Model uses CM technique. We need to determine the number of collocation points and their position in the time interval, we will use two collocation points positioned according to **Equidistant Collocation points** by Table 3.1 and **Gaussian Quadrature Points** by Table 3.3. This choice results in two distinct numerical models, which are subsequently used for comparative analysis.

4.3.1 Numerical Simulations and Performance Analysis

Using the parameters from Table 4.1, we simulate a time response for the position q when a step input $v_{in} = 50[v]$ is applied. The resulting transient responses for each model are shown in Figure 4.4. Visual inspection reveals that the discrete-time gradient model with simplified system matrices exhibits the poorest performance under variations of the sampling time h.

The remaining models demonstrate comparable performance with no appreciable differences. Further order-of-accuracy analysis, presented in Figure 4.3, shows that the Gauss Collocation Points model presents the higher order of accuracy according to the Theory.

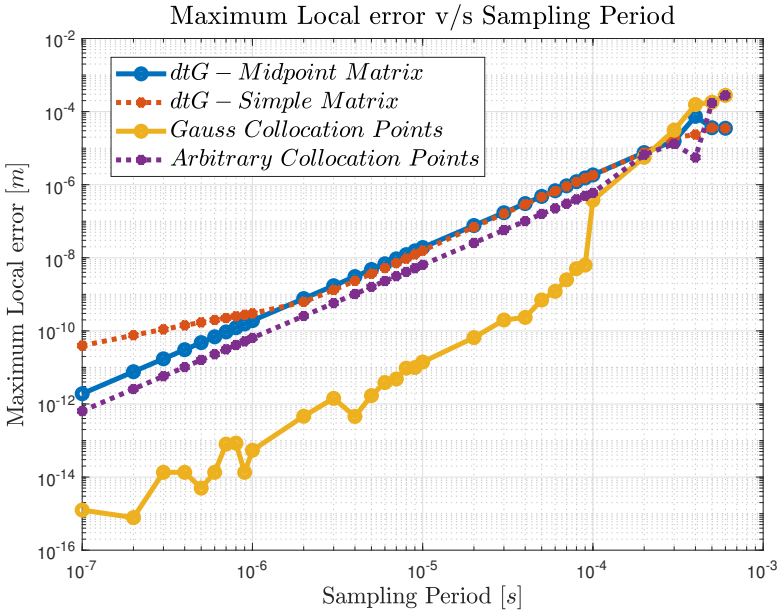


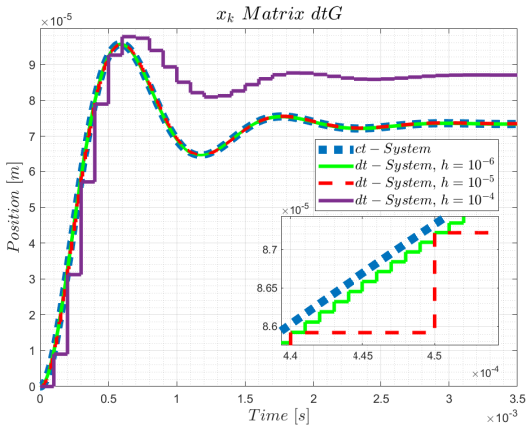
Figure 4.3: Order Graphics

4.4 Position Control

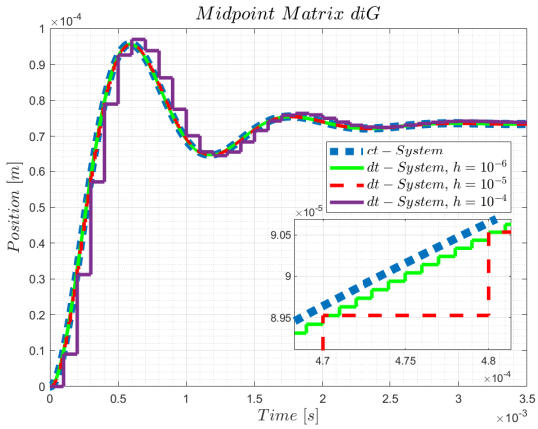
The main applications of PZAs involve precise positioning, which requires control systems specifically designed to accurately regulate their position. Two controller are developed using the methodology presented in Sections 2.5.1 and 2.5.2.

4.4.1 Passivity-Based Controller

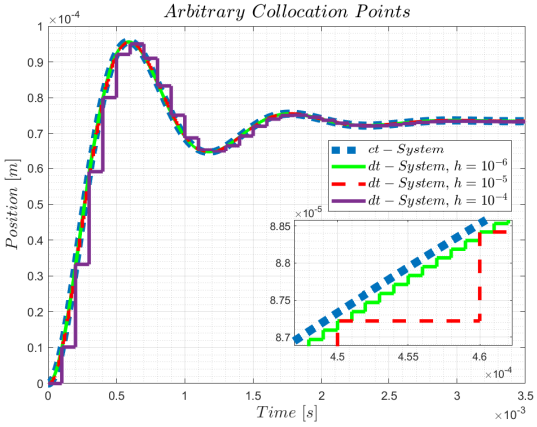
To control the position, it is necessary to shift the minimum energy point from the origin to an admissible equilibrium point (4.3). To do this, we will use ES, we will start by searching for compatible energy functions $H_a(x)$ through the energy shaping matching equation (2.42),



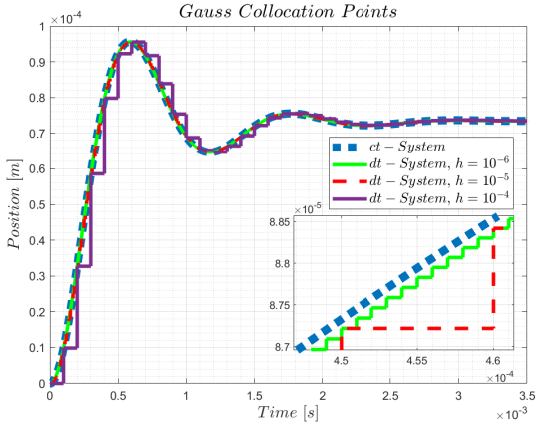
(a) Discrete Gradient approximation using simplified system matrices



(b) Discrete Gradient approximation with midpoint matrices



(c) Collocation method with arbitrary point



(d) Collocation method with Gauss-Legendre points

Figure 4.4: Step response comparison of the four numerical models under study.

where:

$$\nabla H_a(x) = \begin{bmatrix} \nabla_{Q_1} H_a(x) \\ \nabla_{Q_2} H_a(x) \\ \nabla_q H_a(x) \\ \nabla_p H_a(x) \end{bmatrix}$$

$$g^{\perp} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

This yields the following PDE:

$$-h_1^{-1}\left(\frac{Q_1}{C_1}\right)\frac{C_1}{Q_1}\nabla_{Q_1}H_a(x)+\alpha\nabla_p H_a(x)=0 \tag{4.4}$$

$$-h_2^{-1}\left(\frac{Q_2}{C_2}\right)\frac{C_2}{Q_2}\nabla_{Q_2}H_a(x)+\alpha\nabla_p H_a(x)=0 \tag{4.5}$$

$$\nabla_p H_a(x)=0 \tag{4.6}$$

From (4.6), it is observed that $H_a(x)$ does not depend on the momenta p. Substituting this result into equations (4.4) and (4.5), both reduce to the same condition as (4.6). Consequently, $H_a(x)$ is independent of Q_1 and Q_2, leading to:

$$H_a(x)=\Phi(q)$$

Once the form of $H_a(x)$ has been obtained, we propose an energy function for the closed loop. To this end, we recall our control objective, which is to change the position of the PZA q. Accordingly, we shift the desired minimum energy point to a reference value q_{ref}:

$$H_d(x)=\frac{1}{2}\left(\frac{Q_1^2}{C_1}+\frac{Q_2^2}{C_2}+\delta(q-q_{ref})^2+\frac{p^2}{m}\right)$$

where δ is a gain that defines the desired convergence rate. In order to apply the control strategy, we decompose H_d, by choosing $\delta=k+c$, with c a design parameter, then we can rewrite the desired energy as:

$$H_d(x)=H(x)+\left(\frac{c}{2}q^2-(k+c)qq_{ref}+(k+c)q_{ref}^2\right)$$

where:

$$H_a(x)=\frac{c}{2}q^2-(k+c)qq_{ref}+(k+c)q_{ref}^2$$

The gradient is then computed as:

$$\nabla H_a(x)=\begin{bmatrix}0\\0\\c(q-q_{ref})-kq_{ref}\\0\end{bmatrix} \tag{4.7}$$

We observe that (4.7) is consistent with the matching equations (4.4), (4.5), and (4.6). The energy shaping control action is given by (2.43):

$$\beta(x)=-\frac{c}{\alpha}(q-q_{ref})+\frac{k}{\alpha}q_{ref}$$

and the control input is applied through (2.5.2), resulting in:

$$u = -\frac{c}{\alpha}(q - q_{ref}) + \frac{k}{\alpha}q_{ref} + v.$$

The resulting closed-loop system is given by:

$$\Sigma_{ES}: \begin{cases} \begin{bmatrix} \dot{Q}_1 \\ \dot{Q}_2 \\ \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} -h_1^{-1}\left(\frac{Q_1}{C_1}\right)\frac{C_1}{Q_1} & 0 & 0 & \alpha \\ 0 & -h_2^{-1}\left(\frac{Q_2}{C_2}\right)\frac{C_2}{Q_2} & 0 & \alpha \\ 0 & 0 & 0 & 1 \\ -\alpha & -\alpha & -1 & -b \end{bmatrix} \begin{bmatrix} \frac{Q_1}{C_1} \\ \frac{Q_2}{C_2} \\ \left(k + \frac{c}{\alpha}\right)(q - q_{ref}) \\ \frac{p}{m} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \alpha \end{bmatrix} v \\ y = \begin{bmatrix} 0 & 0 & 0 & \alpha \end{bmatrix} \begin{bmatrix} \frac{Q_1}{C_1} \\ \frac{Q_2}{C_2} \\ qk \\ \frac{p}{m} \end{bmatrix} \end{cases}$$

DI control action is applied by feeding back the passive output through the input v as:

$$v = -\gamma\alpha\frac{p}{m}$$

where γ is a design parameter that allows us to adjust the convergence rate. The complete damping injection control law is:

$$v = -\gamma\alpha\frac{p}{m}$$

and the resulting closed-loop system becomes:

$$\begin{bmatrix} \dot{Q}_1 \\ \dot{Q}_2 \\ \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} -h_1^{-1}\left(\frac{Q_1}{C_1}\right)\frac{C_1}{Q_1} & 0 & 0 & \alpha \\ 0 & -h_2^{-1}\left(\frac{Q_2}{C_2}\right)\frac{C_2}{Q_2} & 0 & \alpha \\ 0 & 0 & 0 & 1 \\ -\alpha & -\alpha & -1 & -(b + \gamma\alpha) \end{bmatrix} \begin{bmatrix} \frac{Q_1}{C_1} \\ \frac{Q_2}{C_2} \\ \left(k + \frac{c}{\alpha}\right)(q - q_{ref}) \\ \frac{p}{m} \end{bmatrix} \tag{4.8}$$

The gain γ is selected by analyzing the stability of (4.8). The time derivative of the energy yields:

$$\dot{H}_d(x) = -h_1^{-1}\left(\frac{Q_1}{C_1}\right)\frac{Q_1}{C_1} - h_2^{-1}\left(\frac{Q_2}{C_2}\right)\frac{Q_2}{C_2} - (b + \gamma\alpha)\frac{p^2}{m^2}$$

To ensure stability of the system, γ must satisfy:

$$\gamma > -\frac{b}{\alpha}.$$

Finally, the control input that allows us to regulate the position q is given by:

$$u = -\frac{c}{\alpha}(q - q_{ref}) + \frac{k}{\alpha}q_{ref} - \gamma\alpha\frac{p}{m} \tag{4.9}$$

Analysing (4.9) we can see that the obtained controller is equivalent to a Proportional-Derivative controller with a feedforward term described by

$$PDff = K_P(r - y) + K_D\,\dot{y} + K_{ff}\,r$$

due to

$$\dot{y} = p$$

and where

$$K_P = \frac{c}{\alpha}, \qquad K_D = \gamma\alpha, \qquad K_{ff} = \frac{k}{\alpha\, m}.$$

The equilibrium point of the closed-loop system is:

$$\begin{aligned} h_1^{-1}\left(\frac{Q_1^*}{C_1}\right) &= 0 \\ h_2^{-1}\left(\frac{Q_2^*}{C_1}\right) &= 0 \\ p^* &= 0 \\ q^* &= q_{ref} - \frac{\alpha}{L}\left(\frac{Q_1^*}{C_1} + \frac{Q_2^*}{C_2}\right) \end{aligned}$$

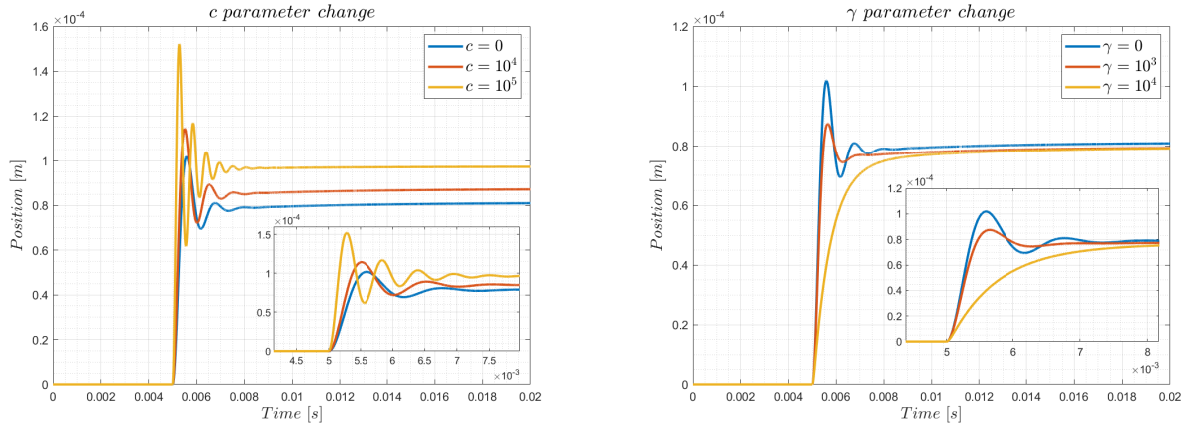
where $L = \left(k + \frac{c}{\alpha}\right)$. It can be seen that the proposed controller does not reach the desired value q_{ref}. The charge in the hysteron does not converge to zero, resulting in an offset with respect to the reference position. Step responses were conducted for different values of c and γ, with $q_{ref} = 10^{-4}\mu(t - 0.005)[m]$ in Figure 4.5.

4.4.2 Proportional-Integral Controller

A PI controller is proposed based on a simplified version of the linear PZA pHs model where the nonlinear admittances $h_i^{-1}(v_i)$ are replaced by the linear admittances slope as

$$h_i^{-1}(v_i) = \rho_i^{-1}v_i$$

Following [34], the simplification assumes that the hysteron dynamics are significantly faster than the mechanical dynamics, allowing us to consider Q_1 and Q_2 as time-invariant. As a



(a) Time response for different c, setting $\gamma = 0$

(b) Time response for different γ, setting $c = 0$

Figure 4.5: Time response for different controller parameters

result, we have:

$$\dot{Q}_1 = 0 = \frac{\rho_1^{-1}}{C_1} Q_1 + \frac{\alpha}{m} p$$
$$\dot{Q}_2 = 0 = \frac{\rho_2^{-1}}{C_2} Q_2 + \frac{\alpha}{m} p$$

Implying that Q_1 and Q_2 are scalar function of the momenta p, then the values of the hysterons states can be replaced into the $\dot{p}$ dynamical equation leading to

$$\begin{aligned} \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} &= \begin{bmatrix} 0 & \frac{1}{m} \\ -k & -\frac{b_{eq}}{m} \end{bmatrix} \begin{bmatrix} q \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha \end{bmatrix} v_{in} \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} q \\ p \end{bmatrix} \end{aligned} \tag{4.10}$$

with

$$b_{eq} = \alpha^2 \left(\frac{1}{\rho_1^{-1}} + \frac{1}{\rho_2^{-1}} \right) + b$$

The gains of the PI controller will be determined wih Pole Placement from Section 2.5.1. Applying (2.14) to (4.10) generates the following transfer function

$$G(s) = \frac{\frac{\alpha}{m}}{s^2 + \frac{b_{eq}}{m} s + \frac{k}{m}} \tag{4.11}$$

Using (2.33) as controller, the A_{cl} with (4.11) is computed

$$A_{cl} = s^3 + \frac{b_{eq}}{m}s^2 + \left(\frac{k + \alpha K_P}{m}\right)s + \frac{\alpha}{m}K_I \tag{4.12}$$

we can see that the controller gains can only change the coefficient of s^1 and s^0 being unable to adjust the coefficient of s^2 implying that the poles cannot be placed freely. Let's consider a target closed-loop gain composed by a simple pole and a complex one:

$$A_{cl}^t = (s + a)\left(s^2 + 2\xi\omega_n s + \omega_n^2\right)$$

with a and ω_n parameters to determine and $\xi = 0.707$ the damping factor. This polynomial must match with the closed-loop gain (4.12), this implies that

$$s^3 + (1.414\omega_n + a)\,s^2 + \left(\omega_n^2 + 1.414\omega_n a\right)s + \omega_n^2 a = s^3 + \frac{b_{eq}}{m}s^2 + \left(\frac{k + \alpha K_P}{m}\right)s + \frac{\alpha}{m}K_I$$

For both equations to be equivalent, it is sufficient for their coefficients to match. It can be observed that there is a constraint on the coefficient of s^2:

$$1.414\omega_n + a = \frac{b_{eq}}{m}$$

For the other two coefficient, the controller gains K_P and K_I will be used to match the equations, then we have the following design equations:

$$\begin{aligned} a &= \frac{b_{eq}}{m} - 1.414\omega_n \\ K_P &= \frac{m(\omega_n^2 + 1.414\omega_n a) - k}{\alpha} \\ K_I &= \frac{m\omega_n^2 a}{\alpha} \end{aligned} \tag{4.13}$$

Then we have a free degree in choosing ω_n. To determine the value, we will analyse the poles of the system (4.11):

$$\begin{aligned} p_1 &= -1206 \\ p_2 &= -20811 \end{aligned}$$

We can see that the dominant pole correspond to p_1, we can choose $\omega_n = 3000$ to be the

dominant pole, then

$$\begin{aligned} a &= 17779 \\ K_P &= 1.3191 \times 10^6 \\ K_I &= 3.5062 \times 10^9 \end{aligned} \tag{4.14}$$

As shown in (4.9), the proportional gain increases the convergence rate by effectively enhancing the system's natural stiffness, as illustrated in (4.8). The controller is implemented in the simplified system (4.10) and in the real system (4.1). A step-response is generated using $r = 1 \times 10^{-4}\mu(t)[m]$, obtaining the Figure 4.6.

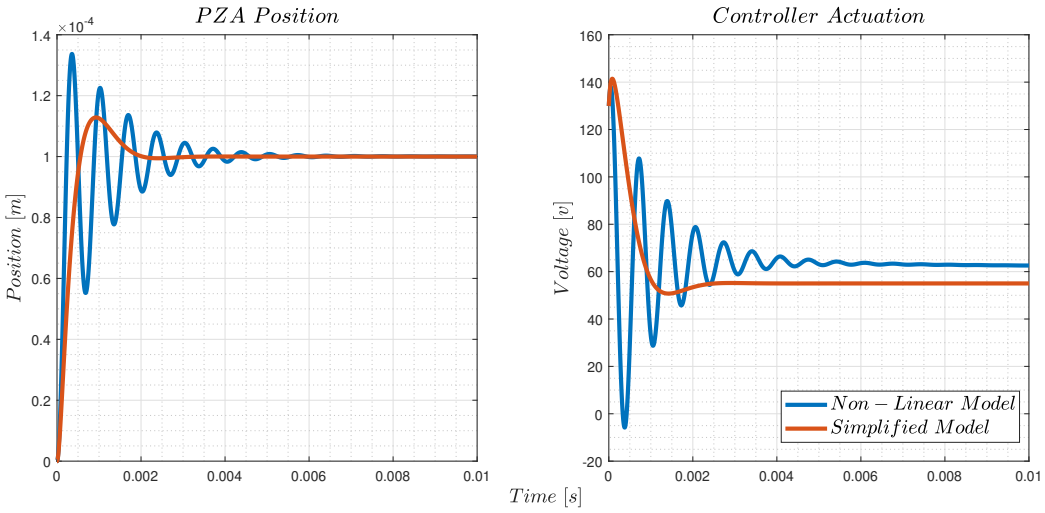


Figure 4.6: Step response with PI controller over the linearized and pHs system

Stability analysis is done using Theorems 2.3.2 and 2.3.3. A linearization of the controlled PZA system (4.5) around the equilibrium point

$$\begin{aligned} -h_1^{-1}\left(\frac{Q_1^*}{C_1}\right) &= 0 \\ -h_2^{-1}\left(\frac{Q_2^*}{C_2}\right) &= 0 \\ p^* &= 0 \\ \xi^* &= \frac{Q_1^*}{K_I\alpha} + \frac{Q_2^*}{K_I\alpha} + \frac{k+\alpha K_P}{\alpha K_I}q^* - \frac{K_P}{K_I}r^* \\ q^* &= r^* \end{aligned} \tag{4.15}$$

is obtained applying (2.8):

$$\begin{bmatrix} \dot{Q}_1 \\ \dot{Q}_2 \\ \dot{q} \\ \dot{p} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \frac{\alpha}{m} & 0 \\ 0 & 0 & 0 & \frac{\alpha}{m} & 0 \\ 0 & 0 & 0 & \frac{1}{m} & 0 \\ -\frac{\alpha}{C_1} & -\frac{\alpha}{C_2} & -k-\alpha K_P & -\frac{b}{m} & \alpha K_I \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta Q_1 \\ \Delta Q_2 \\ \Delta q \\ \Delta p \\ \Delta \xi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \alpha K_P \\ 1 \end{bmatrix} \Delta r$$

Using the parameters from Table 4.1 and (4.14), a transfer function can be obtained using as output equation

$$y = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta Q_1 \\ \Delta Q_2 \\ \Delta q \\ \Delta p \\ \Delta \xi \end{bmatrix}$$

obtaining

$$G(s) = \frac{6.02 \times 10^7 s + 1.6 \times 10^{11}}{s^3 + 3681 s^2 + 9.22 \times 10^7 s + 1.6 \times 10^{11}}$$

The poles of the system are

$$\begin{aligned} s_1 &= -9.4 \times 10^2 + 9.38 \times 10^3 j \\ s_2 &= -9.4 \times 10^2 - 9.38 \times 10^3 j \\ s_3 &= -1.8 \times 10^3 \end{aligned}$$

Finally, the controlled system is stable for any equilibrium point (4.15).

4.5 Controllers Time-Discretization

The PI controller is defined as a state-space system as

$$\begin{aligned} \dot{\xi} &= r - q \\ u &= K_P(r - q) + K_I \xi \end{aligned} \tag{4.16}$$

this controller is discretized with TSS and CM. The target system is computed applying u from (4.16) as input of (4.1), obtaining

$$\begin{bmatrix} \dot{Q}_1 \\ \dot{Q}_2 \\ \dot{q} \\ \dot{p} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} -h_1^{-1}\left(\frac{Q_1}{C_1}\right)\frac{C_1}{Q_1} & 0 & 0 & \alpha & 0 \\ 0 & -h_2^{-1}\left(\frac{Q_2}{C_2}\right)\frac{C_2}{Q_2} & 0 & \alpha & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\alpha & -\alpha & -k-\alpha K_P & -b & \alpha K_I \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{Q_1}{C_1} \\ \frac{Q_2}{C_2} \\ q \\ \frac{p}{m} \\ \xi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \alpha K_P \\ 1 \end{bmatrix} r$$

described in simplified form

$$\dot{x}^{ext} = f(x^{ext}) + B \cdot r \tag{4.17}$$

Then applying CM equations (3.16) on (4.17):

$$\begin{aligned} x_{ci}^{ext} &= x_k^{ext} + h\sum_{j=1}^{s} a_{ij}\left(f(x_{cj}^{ext}) + Br_{cj}\right), \qquad i = 1, ..., s \\ x_{k+1}^{ext} &= x_k^{ext} + h\sum_{i=1}^{s} b_i\left(f(x_{ci}^{ext}) + Br_{ci}\right). \end{aligned} \tag{4.18}$$

With this last system the dynamic of the controller is implemented implicitly, we only need the value of the plant and controller states at the current time, which implies that we need to measure or estimate Q_1, Q_2, q and p at the current time, the controller state at the current time is calculated in the previous iteration. The controllers output doesn't present any change. In the ideal case, we have access to all the state values and the control algorithm is apply, a simulation of the ideal case is performed in Figure 4.7.

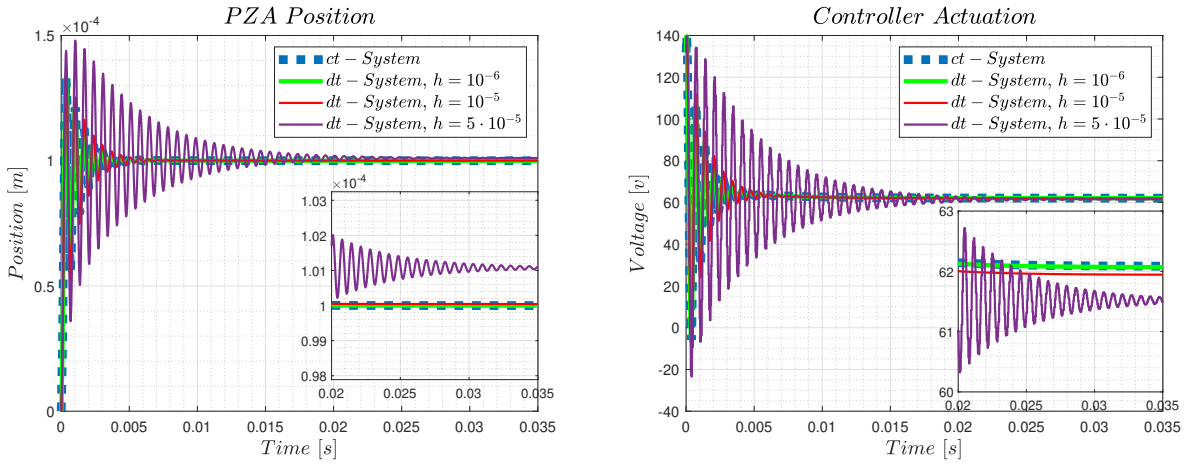


Figure 4.7: PI Time-response

When full state measurement is unavailable, state observers become essential for esti-

mating the required states to implement TSS. However, the system's inherent hysteresis effects difficult the creation of an observer. The deadzone function $h_i^{-1}(\cdot)$ exhibits three possible values, resulting in a nonlinear system that switches between multiple states. Among these, all states in which either of the two $h_i^{-1}(\cdot)$ values equals zero lead to an unobservable system. To overcome this limitation, we propose a simplified linear system based only on the PZA mechanical dynamics. This reduced-order system approximates the actuator as a mass-spring-damper system:

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{m} \\ -k & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} q \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha \end{bmatrix} V_{in}, \tag{4.19}$$

The linear approximation permits standard Luenberger observer implementation and maintains the dominant mechanical dynamics. Combining the simplified system with controller (4.16) gives us a target system with form

$$\begin{bmatrix} \dot{q} \\ \dot{p} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{m} & 0 \\ -k - \alpha K_P & -\frac{b}{m} & \alpha K_I \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ p \\ \xi \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha K_P \\ 1 \end{bmatrix} r, \tag{4.20}$$

The simplified target system only needs the information of the position q that can be measured and the velocity p that can be estimated using a generic Luenberger observer.

Luenberger Observer Design

A Luenberger observer is a linear system that let us estimate the estates of the real system [4]. The observer is applied by

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$

where $\hat{x}$ is the state estimation and $L = [l_1 \; l_2]^\top$ contains the observer gains. The observer dynamics are generated by the matrix $A - LC$, the poles of the system can be calculated by

$$|sI - (A - LC)| = s^2 + \left(\frac{b}{m} + l_1\right) s + \frac{k + l_1 b + l_2}{m} \tag{4.21}$$

The observer gains l_1 and l_2 are designed using as reference the poles of the simplified system

$$\begin{aligned} p_1 &= -1840.5597 + 4564.3045j \\ p_2 &= -1840.5597 - 4564.3045j, \end{aligned}$$

an choosing as the desired observer's poles two simple poles in

$$\begin{aligned} p_1 &= -2000 \\ p_2 &= -20000 \end{aligned} \tag{4.22}$$

which implies that the desired poles takes the form:

$$|sI - (A - LC)|_{desired} = (s - p_1)(s - p_2) = s^2 - (p_1 + p_2)s + p_1 p_2 \tag{4.23}$$

The observer gains are then computed by matching the coefficients of (4.21) and (4.23), yielding:

$$\begin{aligned} l_1 &= 319, \\ l_2 &= -21412.6564. \end{aligned}$$

Using the observer, the value of p can be estimated, which enables the discretization of

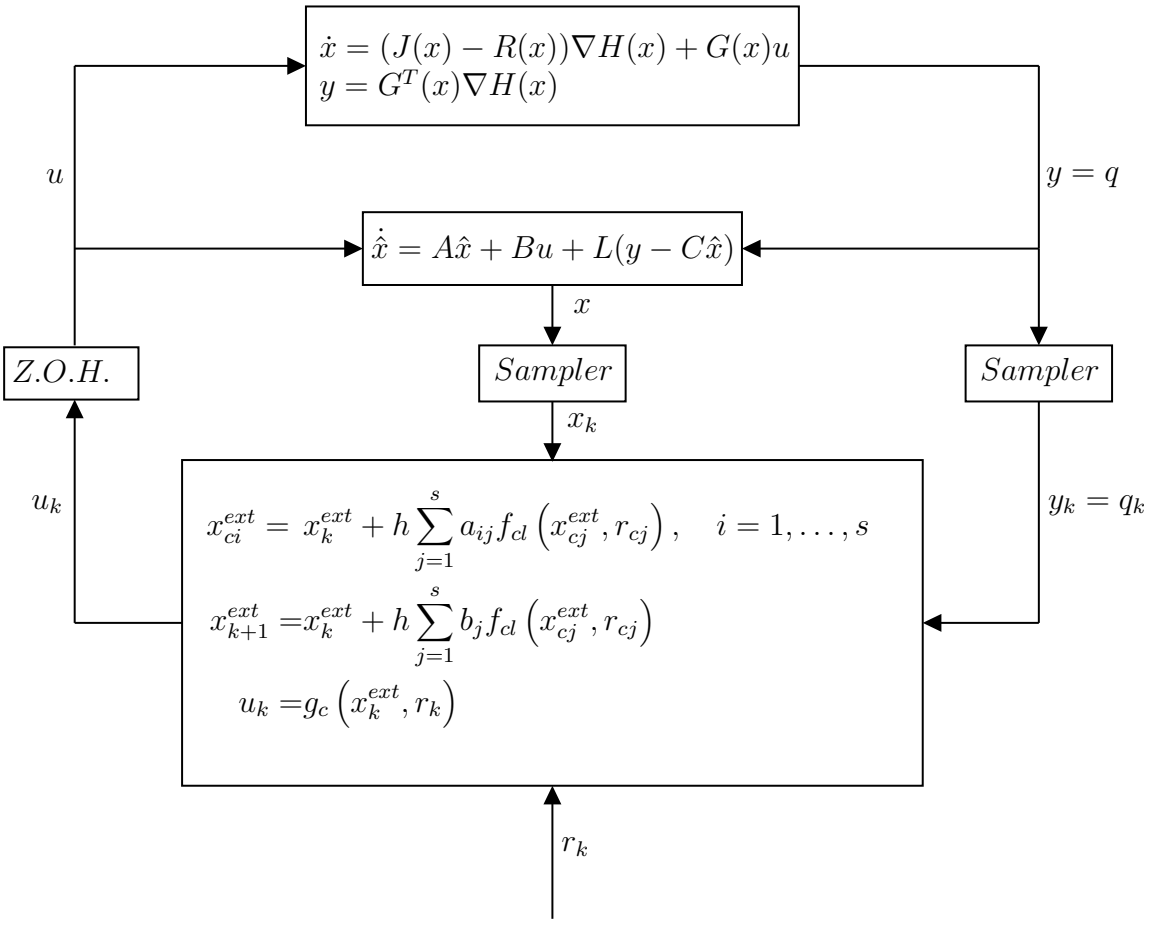


Figure 4.8: Control diagram illustrating the implementation of TSS with a continuous-time observer

equation (4.20) using the CM method. An implementation diagrama is shown in Figure 4.8 and simulations are carried out for different sampling times h of the controller in Figure 4.9. We can see from it that the system doesn't reach the reference signal, using a simplified system for the observer ads errors estimating the states.

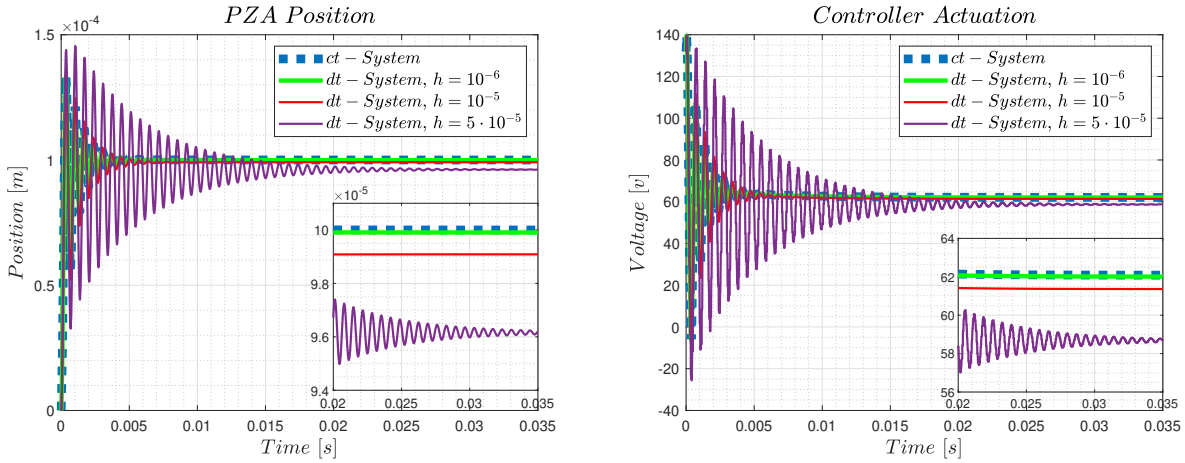


Figure 4.9: PI with Continuous-Time Observer Time-response

Finally, to generate a complete discrete-time control scheme is necessary to obtain a discrete-time observer, using the continuos-time as a base, the discrete-time observer can be generated discretizing the Simplified System (4.19) by ZOH (3.9). The resulting system obtained with sampling time $h = 10^{-5}$

$$\begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} 0.9988 & 0.009671 \\ -0.2412 & 0.9627 \end{bmatrix} \begin{bmatrix} q_k \\ p_k \end{bmatrix} + \begin{bmatrix} 2.254 \cdot 10^{-9} \\ 4.545 \cdot 10^{-7} \end{bmatrix} V_k^{in}.$$

For the observer poles we discretize the continuous-time ones (4.22) by

$$p_1 = e^{-2000 \times 10^{-5}} = 0.9802$$
$$p_2 = e^{-20000 \times 10^{-5}} = 0.8187$$

Finally, the observer gains are

$$l_1 = 0.1626$$
$$l_2 = -0.5020$$

It is important to note that the observer gains depend directly on the sampling time h. An scheme illustrating the implementation is shown in Figure 4.10.

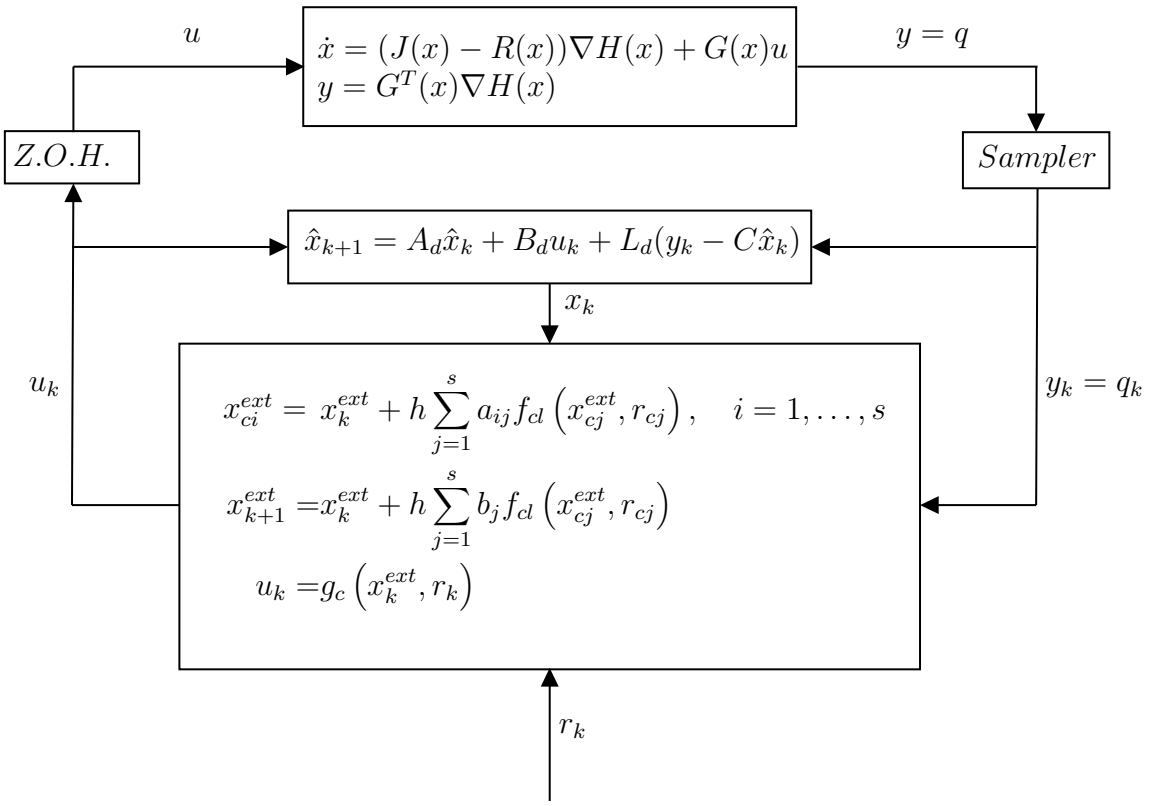
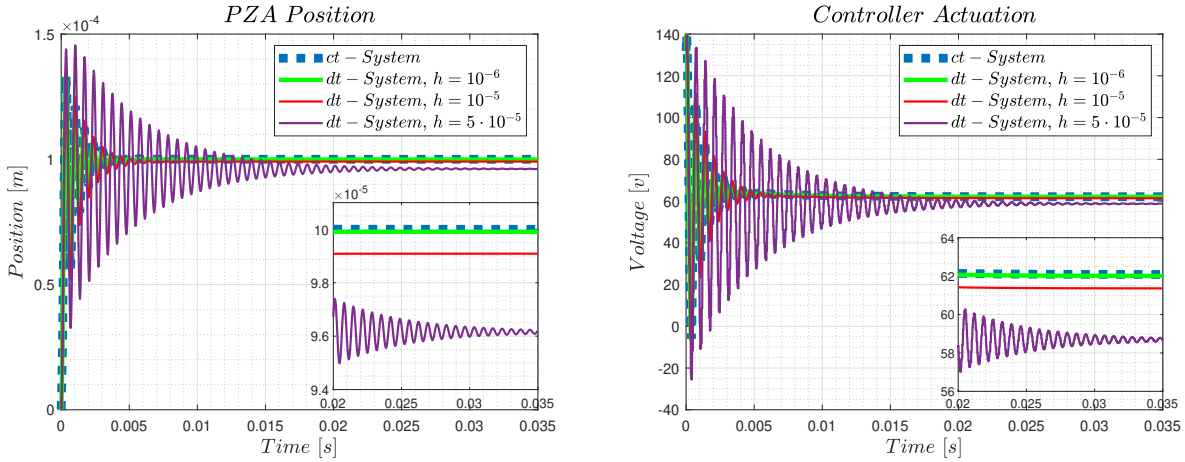


Figure 4.10: Control diagram illustrating the implementation of the target system methodology with a discrete-time observer

Therefore, the previous step response simulation is repeated, this time using the discrete-time observer, resulting in the plot shown in Figure 4.11. Comparing the time response between the discrete and continuous time observer, there are no visible differences.



(a) Controller Actuation

(b) System Response

Figure 4.11: PI with Discrete-Time Observer Time-response

Chapter 5

Experimental Implementation

As final chapter of this Thesis, PI controller designed in the previous chapter will be implemented in a laboratory experiment. For this purpose, the parameters of the PZA to be used will be estimated, and the controller gains will be recalculated accordingly. Finally, the developed controllers will be applied in a real PZA. The performance of the proposed control strategy will be assessed through sampling-time variation tests and comparisons with classical discrete-time controllers.

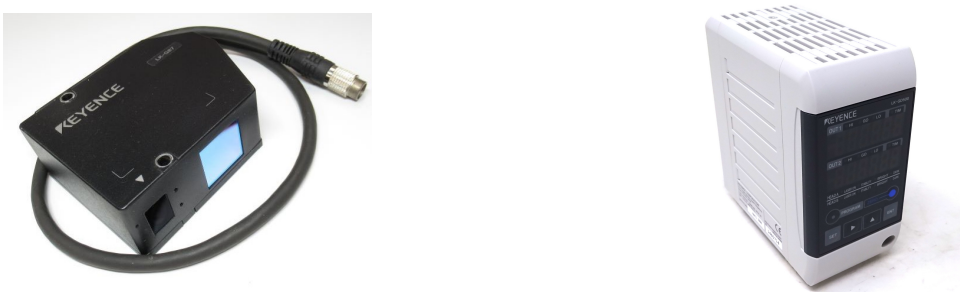
5.1 Experimental Setup

The experimental implementation was conducted using the setup shown in Figure 5.5. The configuration was designed to provide high-precision measurement, low-latency signal processing, and sufficient actuation capabilities for the control of a piezoelectric bender. The main components are described as follows:

1. **dSpace MicroLabBox:** A high-performance, all-in-one real-time control platform, equipped with a dual-core processor running at 2 GHz, an user-programmable FPGA, and more than one hundred high-speed I/O channels. The system supports deterministic execution of control algorithms and offers a wide range of analog and digital interfaces, enabling seamless integration of sensors and actuators.

Figure 5.1: dSpace MicroLabBox

2. **Optical Position Sensor:** Model LK-G87 from KEYENCE, based on a red semiconductor laser ($\lambda = 655\,\text{nm}$) that measures displacement relative to an 80 mm reference point. The sensor provides a measurement range of $\pm 15\,\text{mm}$ with a minimum sampling period of $20\,\mu\text{s}$, ensuring high temporal resolution and low measurement noise. To connect the sensor to the MicroLabBox, the LK-G3001PV controller is used.



(a) LK-G87 **(b)** LK-G3001PV

Figure 5.2: Optical position sensor and controller.

3. **Linear Amplifier:** Pendulum Instruments Dual Channel High Voltage Linear Amplifier A400DI. This device integrates two independent A400 amplifier channels sharing a common ground and power supply. It is used to drive the piezoelectric actuator, providing output voltages up to 100 V with low distortion, enabling precise positioning control.

Figure 5.3: A400DI linear amplifier.

4. **Piezoelectric Bender:** ThorLabs PB4NB2S, capable of producing a maximum displacement of $\pm 450\,\mu\text{m} \pm 15\%$ at an applied voltage of 150 V. The actuator exhibits less than 15% hysteresis and has a free length of 28 mm, making it suitable for high-bandwidth precision motion applications.

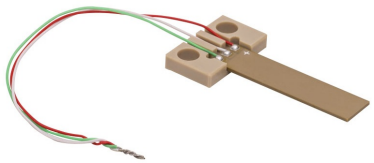


Figure 5.4: ThorLabs PB4NB2S piezoelectric bender.

5. **Computer:** Used to program and control the dSpace MicroLabBox.

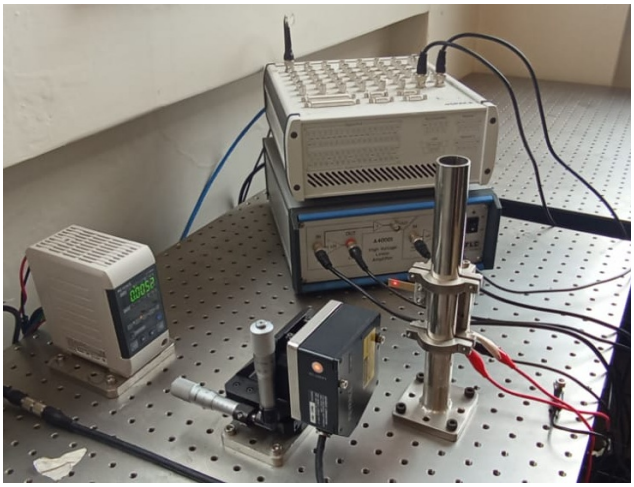


Figure 5.5: Experimental setup for PI Controller implementation.

On the software side, the dSpace MicroLabBox is programmed in Simulink (MATLAB) as a block diagram, which is then compiled and deployed to the platform. Communication between the PC and the MicroLabBox is managed through *ControlDesk*, a software tool provided by dSpace. This interface allows reading and writing of variables in real time, as well as monitoring and adjusting parameters from the connected sensors and actuators.

5.2 Estimation of Parameters

For estimating the parameters of the PZA, we will employ the methodology outlined in [27] (**Section 4.2**). To do this, an experiment will be conducted where the PZA will be excited with a sinusoidal signal of $1[Hz]$ frequency and $55[V]$ amplitude. Following this, measurements of the applied voltage and the displacement of the PZA will be taken. These measurements will be saved using the *iddata* command, which will generate a data object necessary for utilizing the commands in the *System Identification Toolbox*. Once the data file has been created, a *Gray-box System* object will be created using the command *idnlgrey*, transforming the system of equations that model the PZA into an *idnlgrey* variable. This variable, along with the *iddata* variable, will then be used as inputs to the *nlgreyest* command to identify the parameters of the system. The optimization command needs a initial condition to start the optimization proces, we will use the parameters from Table 4.1, the identified parameters are shown in the following table. With these parameters we obtain a 92.9% fit of estimation data.

Parameter	Value	Units
m	1.0148×10^{-3}	$[kg]$
k	25479	$[N/m]$
b	3.7356	$[Ns/m]$
C_1	0.001	$[F]$
ρ_1^{-1}	0.6002	$[\Omega^{-1}]$
d_1	14.7571	$[V]$
C_2	1.9022×10^{-7}	$[F]$
ρ_2^{-1}	2.703×10^{-4}	$[\Omega^{-1}]$
d_2	8.6838	$[V]$
α	0.0774	$[C/m]$

Table 5.1: Parameters for the PHS model obtained with the identification process using sinusoidal signal of $1[Hz]$ frequency and $55[v]$ amplitude.

Parameter Validation for Sinusoidal Input

Sinusoidal responses were applied to both the pHs model and the experimental system at four excitation frequencies, $1[Hz]$, $10[Hz]$, $50[Hz]$ and $100[Hz]$, all of them with amplitude of $55[V]$. The corresponding time-domain responses are depicted in Figures 5.6 to 5.9, while the hysteresis curves are shown in Figures 5.10 to 5.13.

The results indicate that the pHs model captures the overall dynamics of the real system with good agreement at low frequencies. The main discrepancies arise at the response peaks, where the model tends to deviate from the experimental measurements. Regarding the hysteresis behaviour, the model provides an accurate approximation for the $1[Hz]$ input. However, for excitation frequencies of $10[Hz]$ and above, the model's hysteresis curve increasingly deviates from the experimental one.

Parameter validation for pulse input

Since the control objective is to regulate the PZA position to a constant reference, it is necessary to validate that the identified parameters correctly capture the system dynamics when subjected to step references. However, such references may damage the PZA structure; therefore, the input signal is filtered through a system that removes high-frequency components, described by the transfer function

$$F(s) = \frac{1000}{s + 1000}.$$

It should be noted that the developed model does not account for vibration and creep effects [27]; hence, it cannot accurately reproduce highly oscillatory phenomena. This limi-

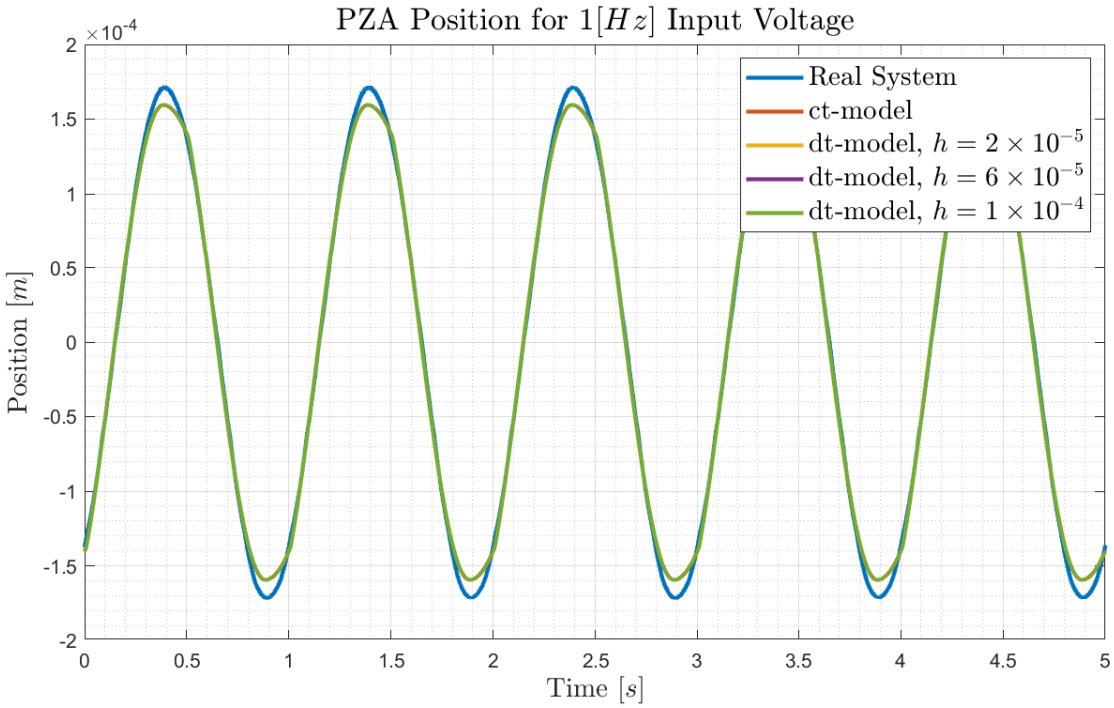


Figure 5.6: Time-Response for $1[Hz]$ Input Voltage

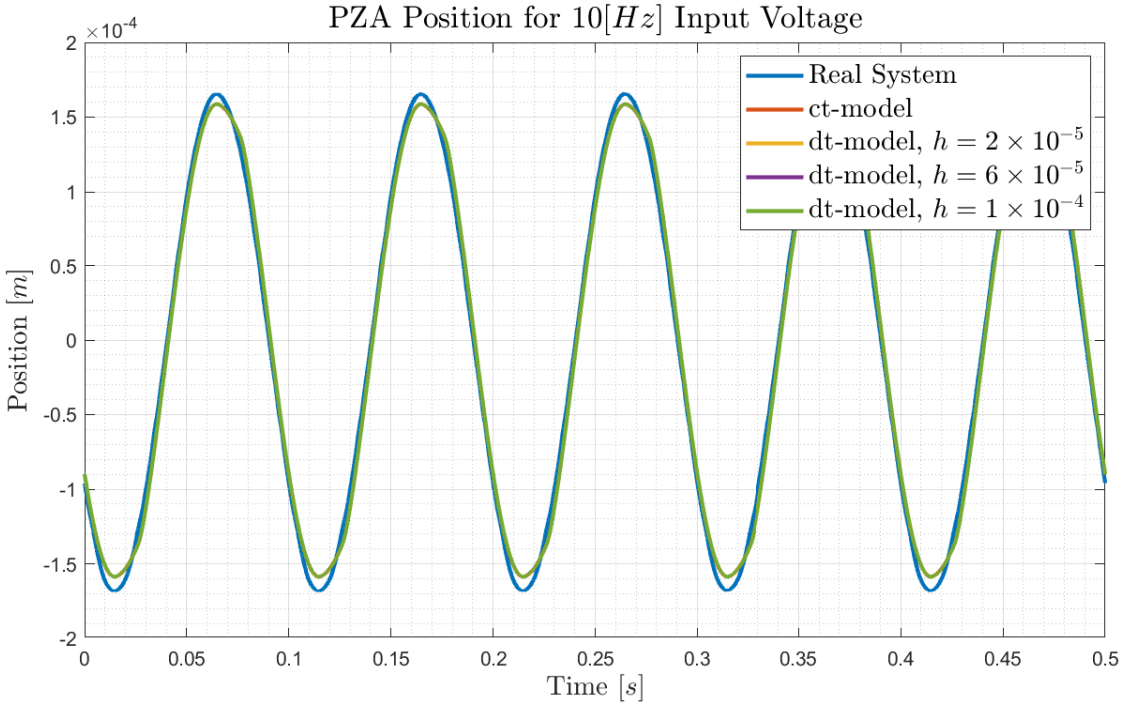


Figure 5.7: Time-Response for $10[Hz]$ Input Voltage

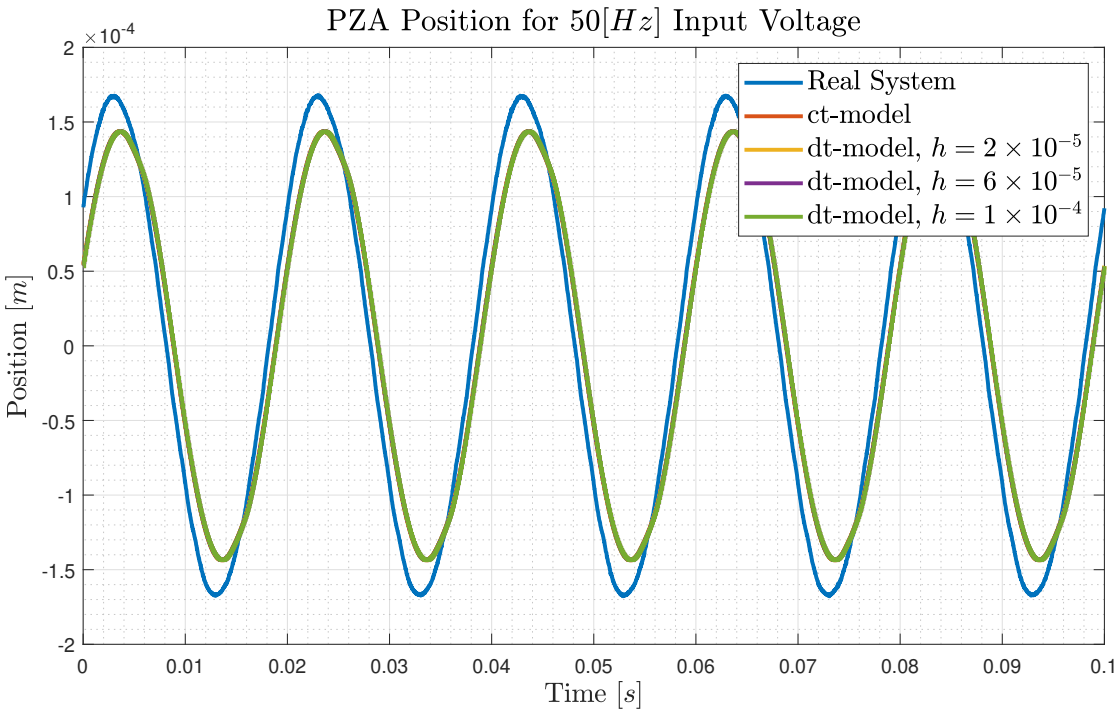


Figure 5.8: Time-Response for $50[Hz]$ Input Voltage

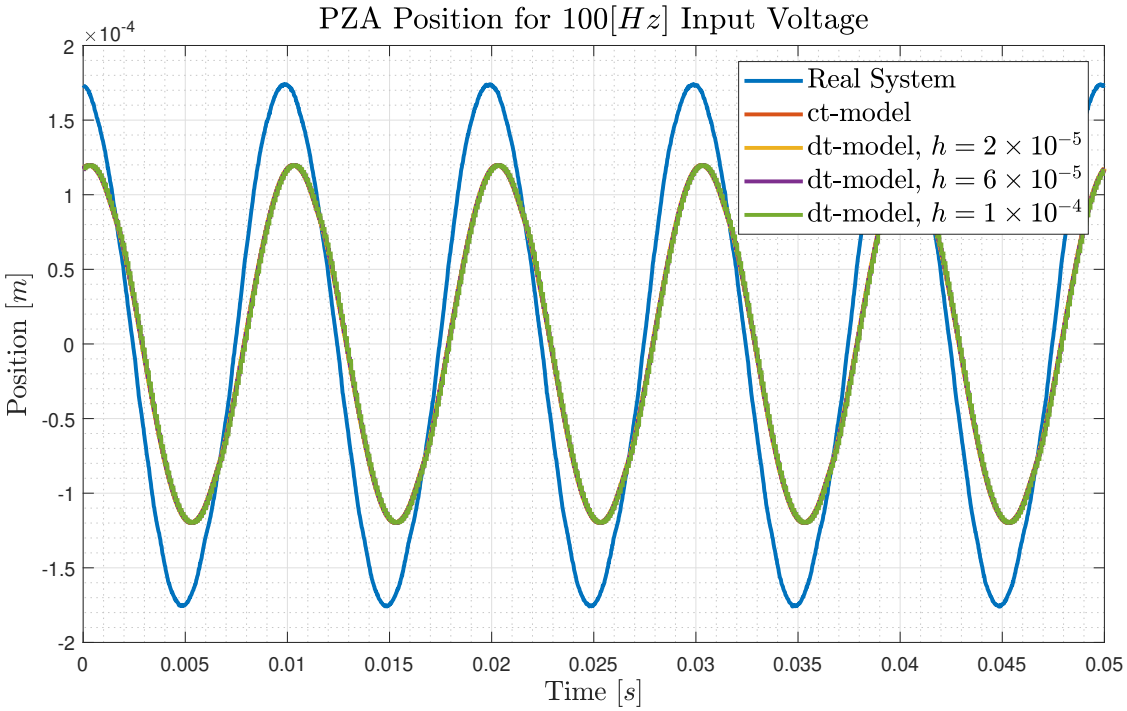


Figure 5.9: Time-Response for $100[Hz]$ Input Voltage

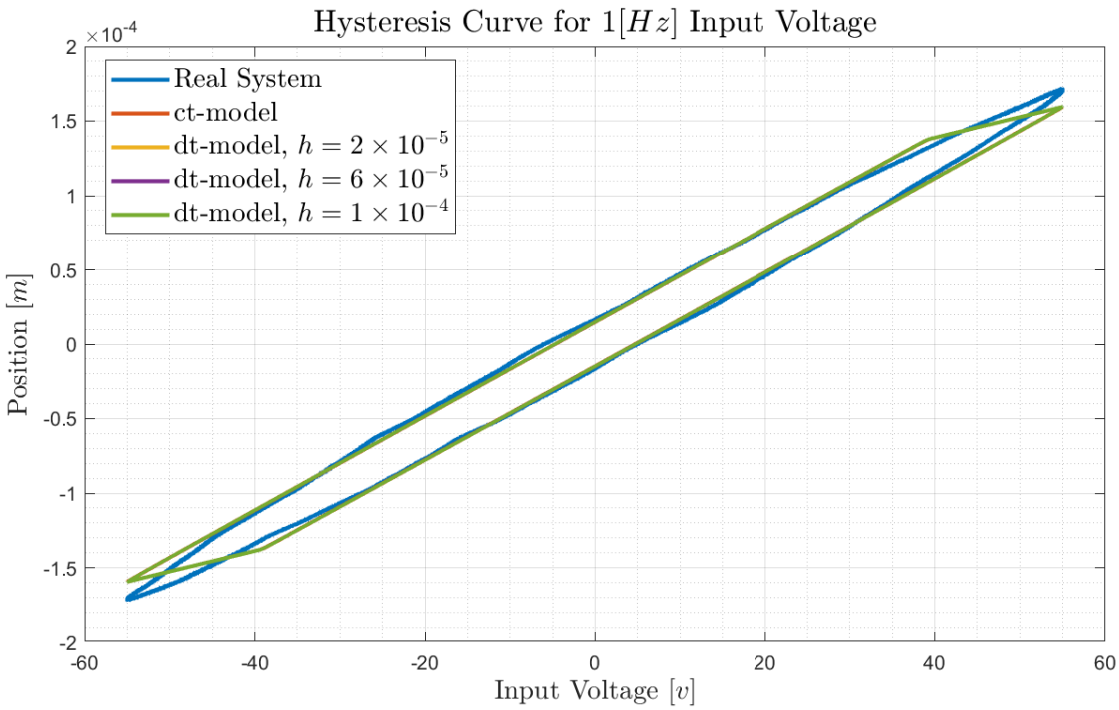


Figure 5.10: Hysteresis Curve for $1[Hz]$ Input Voltage

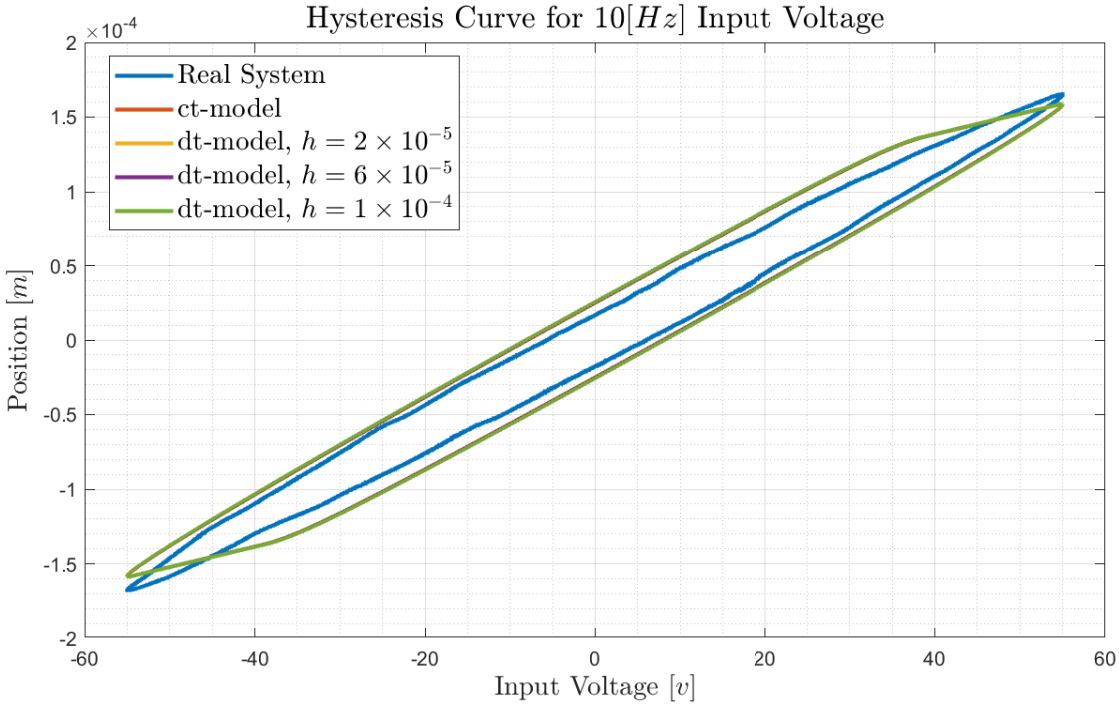


Figure 5.11: Hysteresis Curve for $10[Hz]$ Input Voltage

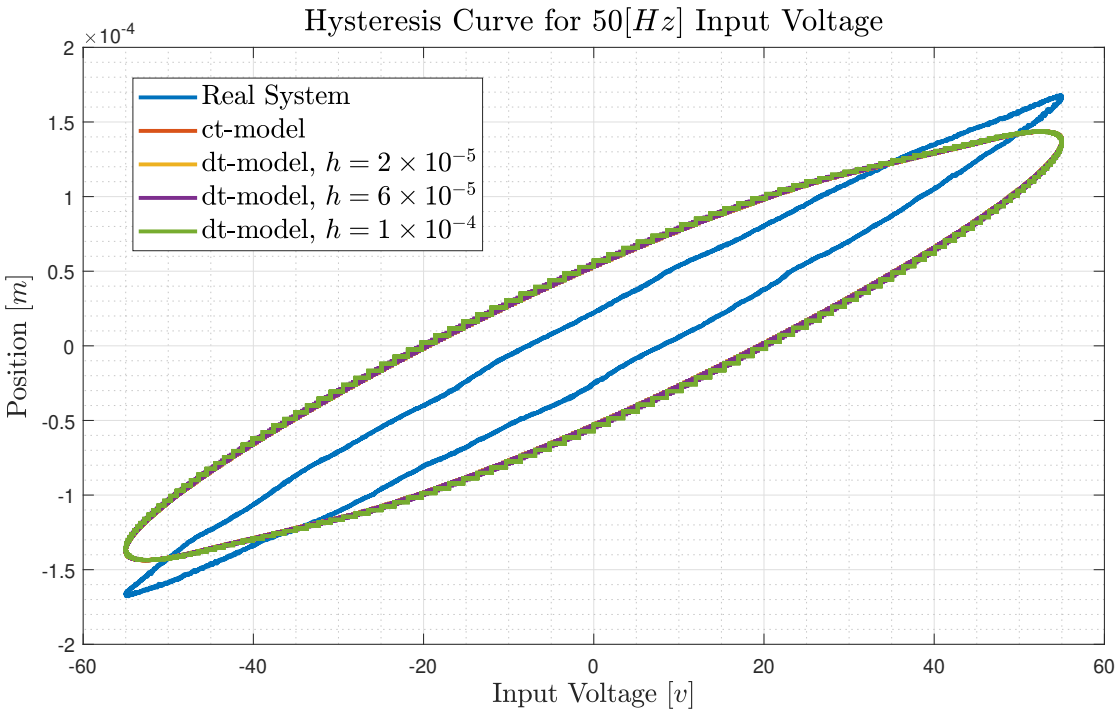


Figure 5.12: Hysteresis Curve for 50[Hz] Input Voltage

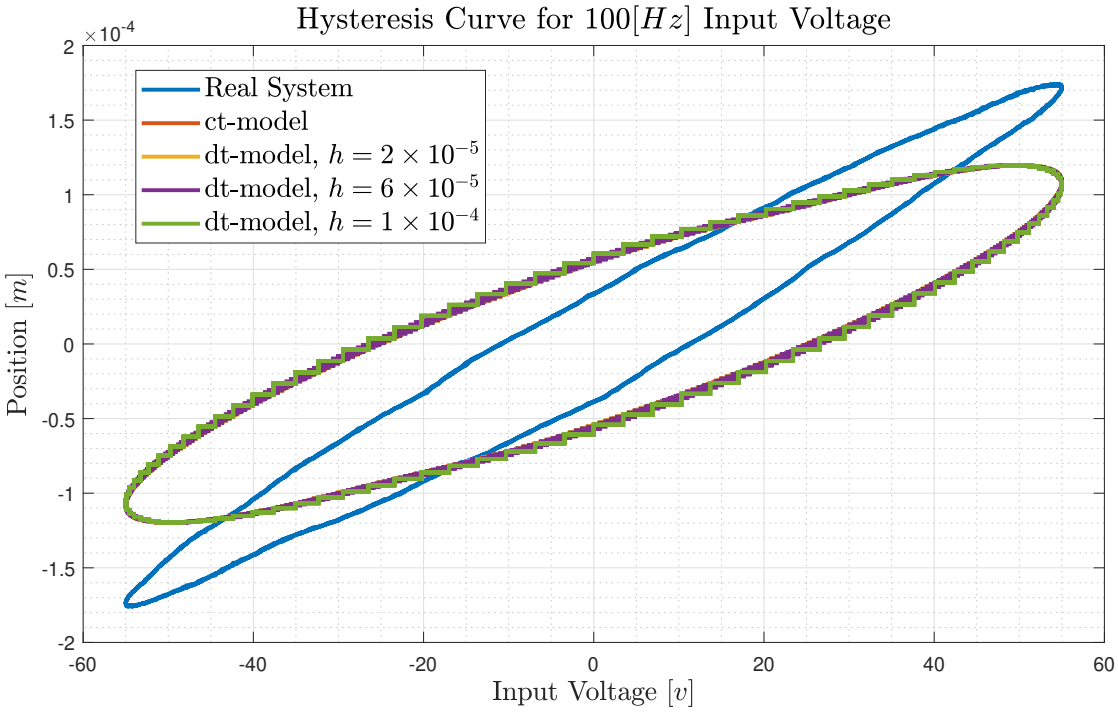


Figure 5.13: Hysteresis Curve for 100[Hz] Input Voltage

tation is illustrated in Figure 5.14, where a pulse train with a period of $1[s]$ and amplitude of $50[v]$ was applied both to the identified model and to the real system. As observed, the model does not capture the existing vibrations and also fails to reach the same amplitude as the real system.

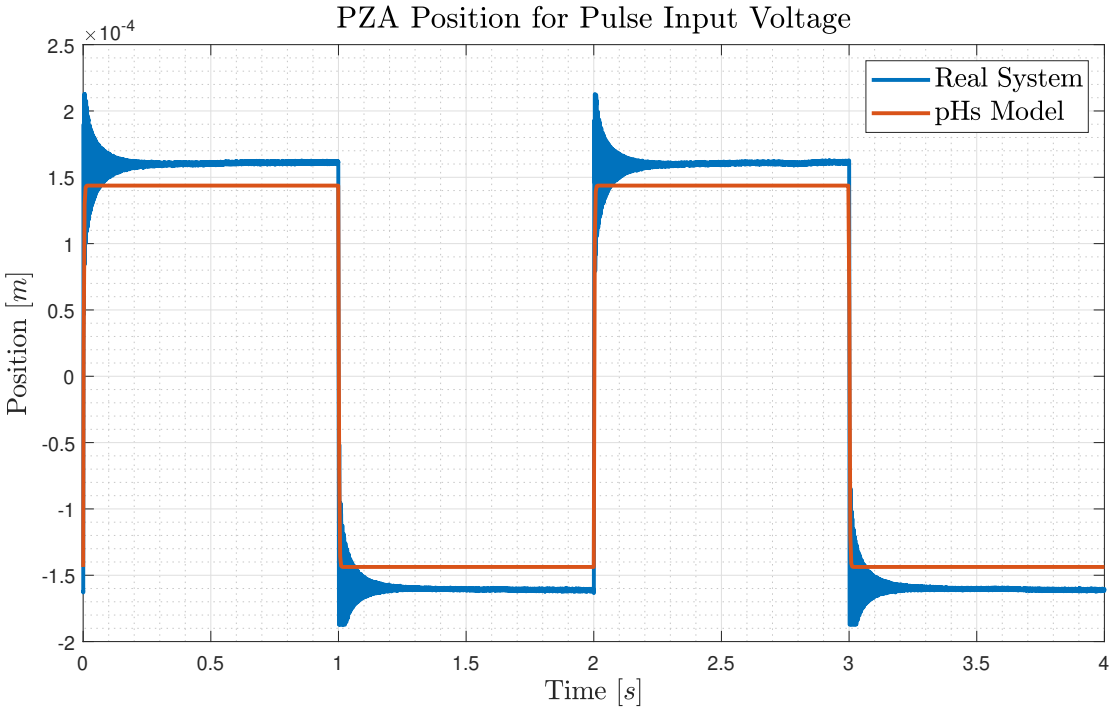


Figure 5.14: System response for a $1[s]$ pulse train input of $50[v]$ amplitude.

Given the discrepancy between the pHs model and the PZA behaviour, a new parameter identification was performed using the same pulse-train reference employed in the previous validation, following the method described earlier. The resulting parameters are summarised in Table 5.2.

Finally, to validate the updated model, a pulse train with a period of $3[s]$ and amplitude of $40[v]$ was applied both to the model and to the real system. The obtained results are shown in Figure 5.15. Again, the oscillatory behaviour is not accurately represented; however, the steady-state error is slightly smaller compared to the previous parameter set.

Controller Parameter Adjustment

Given that the parameters obtained from the experimental model differ from those used in the initial controller design, it is necessary to recalculate the PI controller gains. Using

Parameter	Value	Units
m	1.0148×10^{-3}	$[kg]$
k	25479	$[N/m]$
b	3.7356	$[Ns/m]$
C_1	0.0017	$[F]$
ρ_1^{-1}	0.6002	$[\Omega^{-1}]$
d_1	14.7571	$[V]$
C_2	2.0469×10^{-6}	$[F]$
ρ_2^{-1}	1.9228×10^{-4}	$[\Omega^{-1}]$
d_2	8.6838	$[V]$
α	0.0777	$[C/m]$

Table 5.2: Parameters of the pHs model.

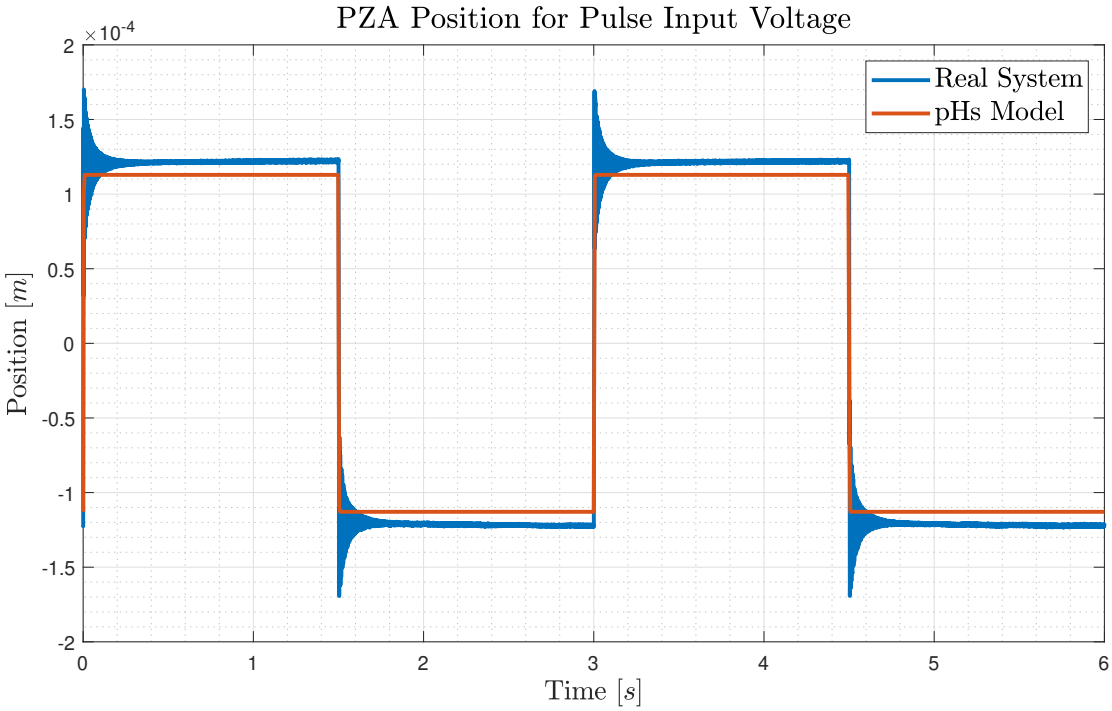


Figure 5.15: System response for a 3[s] pulse train input of 40[v] amplitude.

the parameters from Table 5.2 in (4.13) we compute the controller gains obtaining:

$$\begin{aligned} K_P &= 1.3809 \times 10^6, \\ K_I &= 3.3516 \times 10^9. \end{aligned} \tag{5.1}$$

5.3 Experimental Tests

As final part of the present work, we will realize experimental control test to check the real control behaviour.

5.3.1 General Implementation

To conclude the work presented in this thesis, the developed control system is implemented using the following scheme:

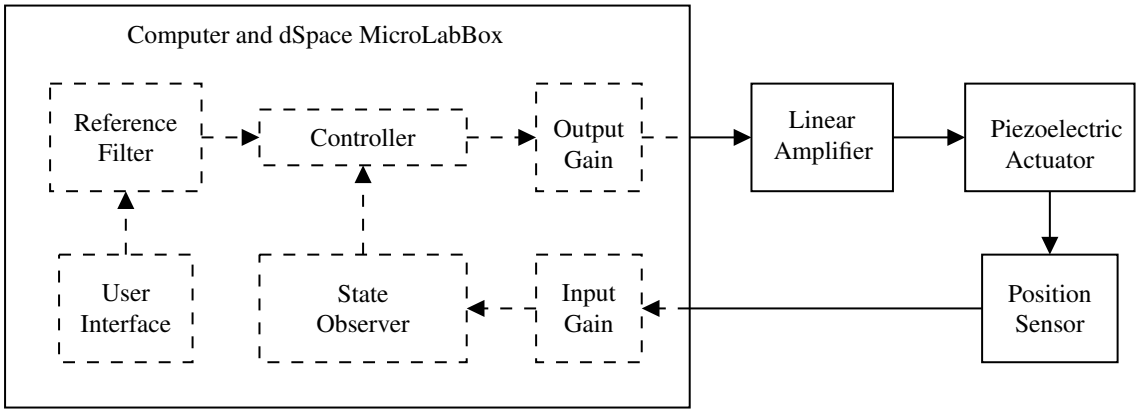


Figure 5.16: General Control Diagram

The control algorithm is structured around six main blocks:

1. **User Interface:** Through Control Desk, the user can monitor the measured position and the applied control action. Additionally, the user can modify the reference signal and various controller parameters.

2. **Reference Filter:** This filter is used to smooth step-type reference inputs to avoid damaging the PZA, the filter dynamic must be fast enough to not affect the dynamic of the controlled system. It has two implementations depending on the type of controller: for continuous-time controllers, the filter is defined by

$$\begin{aligned} \dot{x} &= -1000x + 1000u \\ y &= x \end{aligned},$$

whereas for discrete-time controllers, a Z.O.H. DTA is used:

$$\begin{aligned} x_{k+1} &= e^{-1000h}x_k + \left(1 - e^{-1000h}\right)u_k \\ y_k &= x_k \end{aligned}$$

the discrete-time filter let us consider the reference as a constant value at the collocation points.

3. **Controller:** This block implements the various controllers developed throughout the thesis.

4. **Observer:** State observer required for implementing the DTA using Collocation Methods.

5. **Input and Output Gain:** Input and output gains necessary to correctly scale signals for the complete system.

5.3.2 Noise Analysis and Controller Gain update

The PI controller with gains defined in (5.1) was implemented on the real PZA. The plant became unstable, exhibiting oscillations that exceeded the sensor's measurement range. Analysis of the experimental setup revealed that the position sensor introduced measurement noise. Therefore, the controller gains were recalculated to ensure that the closed-loop configuration did not increase the output noise variance. The position signal was measured in open loop, as shown in Figure 5.17, obtaining the following statistics:

$$\begin{aligned} \text{mean} &= 0, \\ \text{variance} &= 1.2506 \times 10^{-13}. \end{aligned}$$

Starting from the gains in (5.1), they were reduced until the closed-loop noise variance matched the open-loop value. The final gains were:

$$\begin{aligned} K_P &= 10^3, \\ K_I &= 10^7, \end{aligned} \tag{5.2}$$

yielding the following closed-loop noise statistics:

$$\begin{aligned} \text{mean} &= 0, \\ \text{variance} &= 1.0480 \times 10^{-13}. \end{aligned}$$

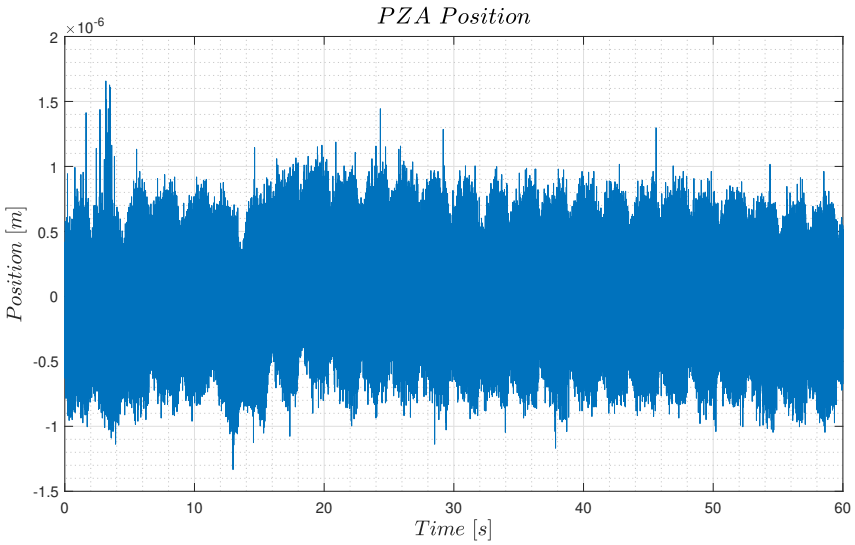


Figure 5.17: Measurement noise.

5.3.3 Test 1: Change of sampling time

As first experimental test, the PI controller with parameters given in (5.2) is implemented in two configurations:

1. **Continuous-time:** The continuous PI controller described in the state-space form (4.16) is implemented using the control loop shown in Figure 2.2.

2. **Discrete-time:** The DTA approximation from TSS (4.20), obtained through the CM method with two Gauss collocation points, is implemented using the control loop shown in Figure 4.10, where the observer gains are recomputed based on the updated parameters in Table 5.1.

A constant-reference tracking test is then performed. Initially, the controller drives the PZA position to zero, after which the reference value is changed to 100$[nm]$. The experimental results are shown in Figures 5.19 and 5.18.

It can be observed that all controllers exhibit similar transient behaviour. However, the discrete-time controllers fail to perfectly reach the reference value, and the steady-state error increases with the sampling period h. This trend matches the simulation results obtained when the controller is implemented with an observer—both in continuous and discrete time—shown in Figures 4.9 and 4.11. Therefore, the use of a simplified observer prevents the achievement of the desired control objective.

5.3.4 Test 2: Forward Euler Comparison

A second test was carried out by comparing the CM discrete-time controller with a forward Euler approximation of the PI controller:

$$\begin{aligned} \xi_{k+1} &= \xi_k + h(r_k - q_k), \\ u_k &= K_I \xi_k + K_P(r_k - q_k). \end{aligned}$$

It is worth noting that, in this case, the forward Euler approximation is equivalent to a ZOH approximation. Using the minimum possible sampling time $h = 2 \cdot 10^{-5}$, a pulse response was obtained and is shown in Figures 5.20 and 5.21. It can be observed that the Euler approximation achieves better performance than the CM controller. This difference in performance is likely due to the simplified observer design used in the CM controller.

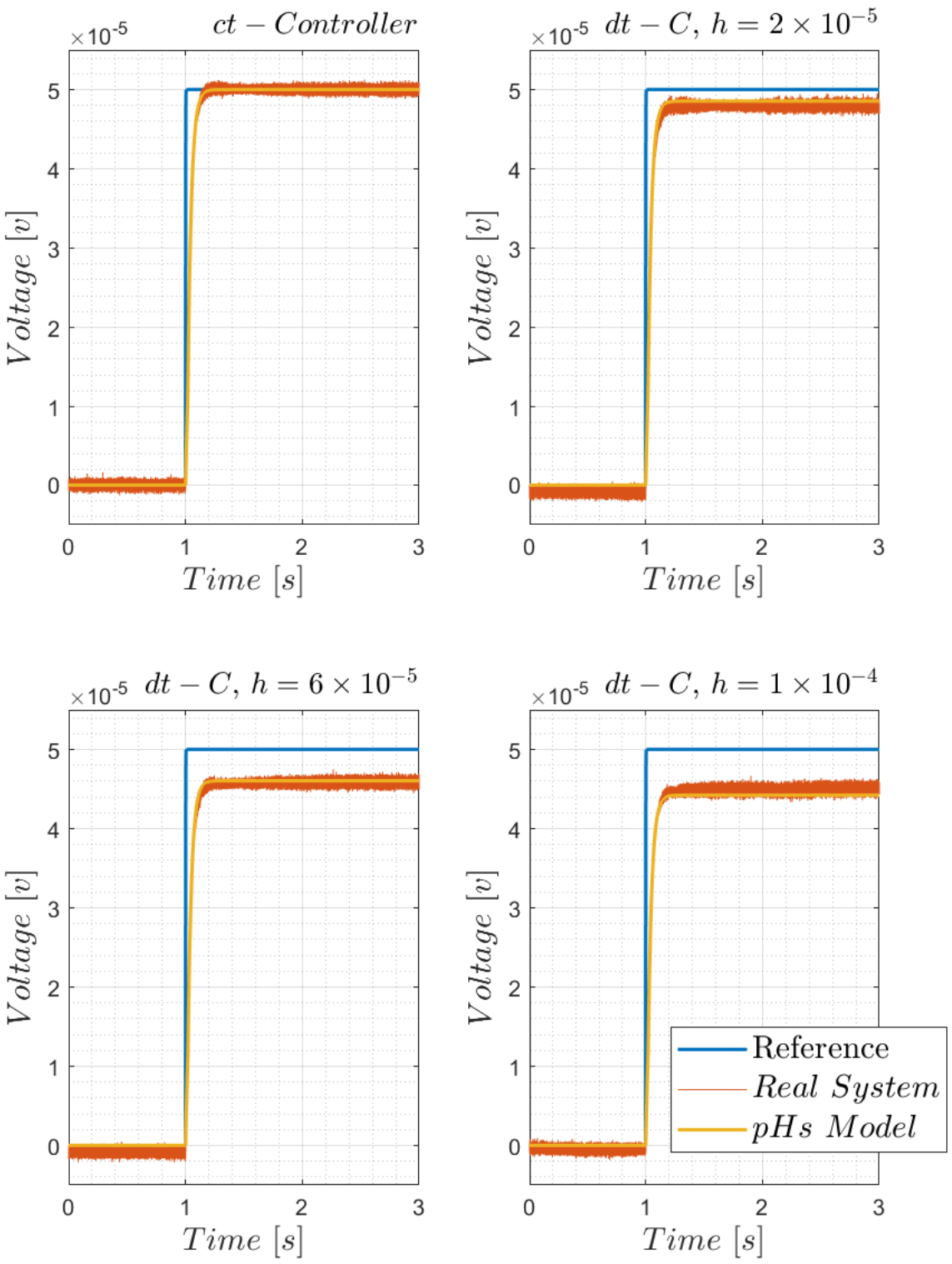


Figure 5.18: Test 1: PZA position

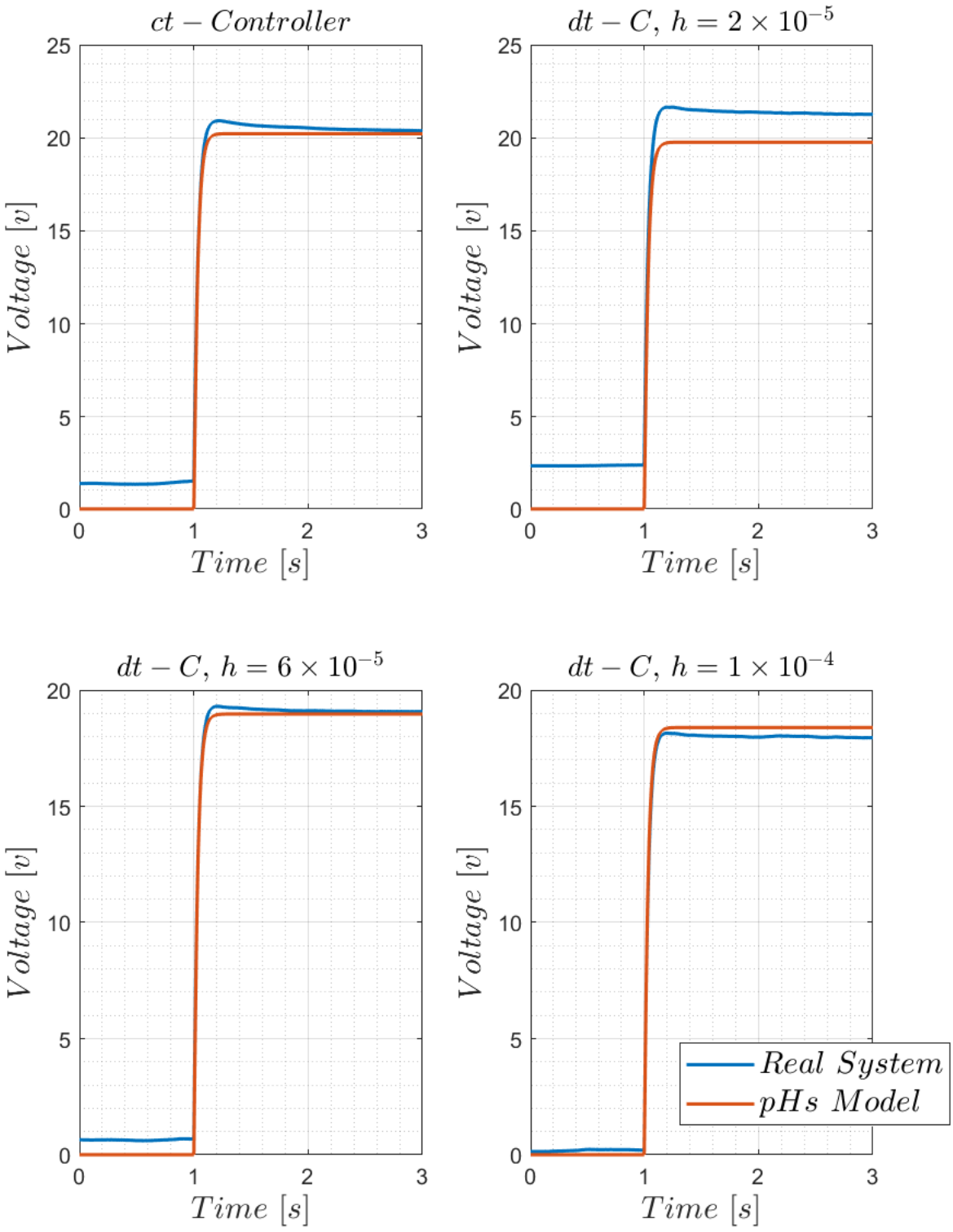


Figure 5.19: Test 1: Control Action

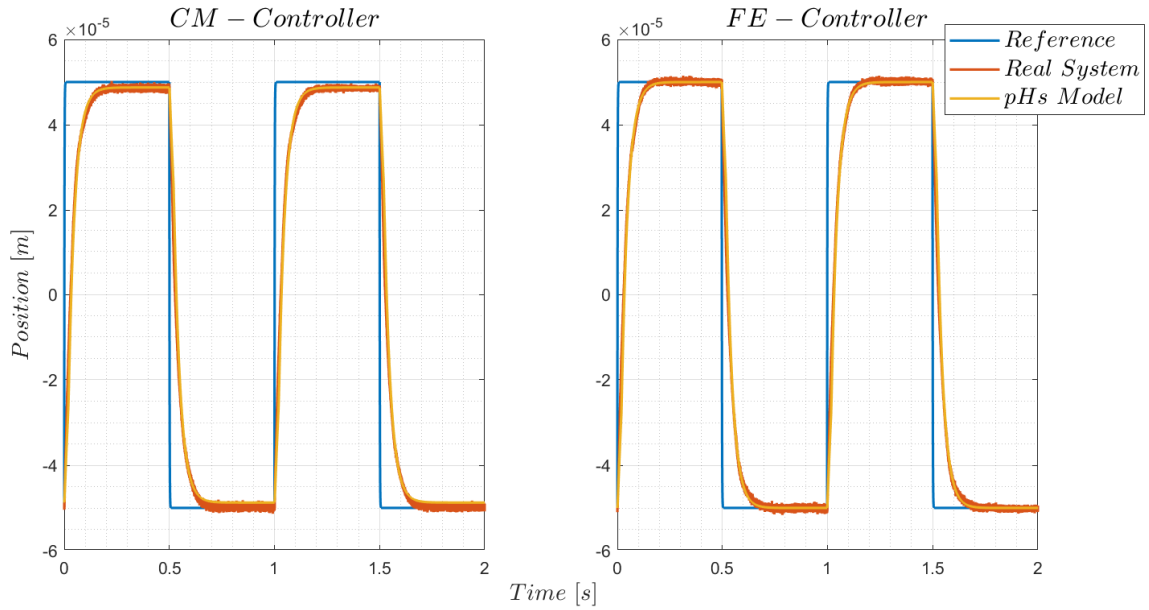


Figure 5.20: Test 2: Position Comparison Between CM and Euler Controllers

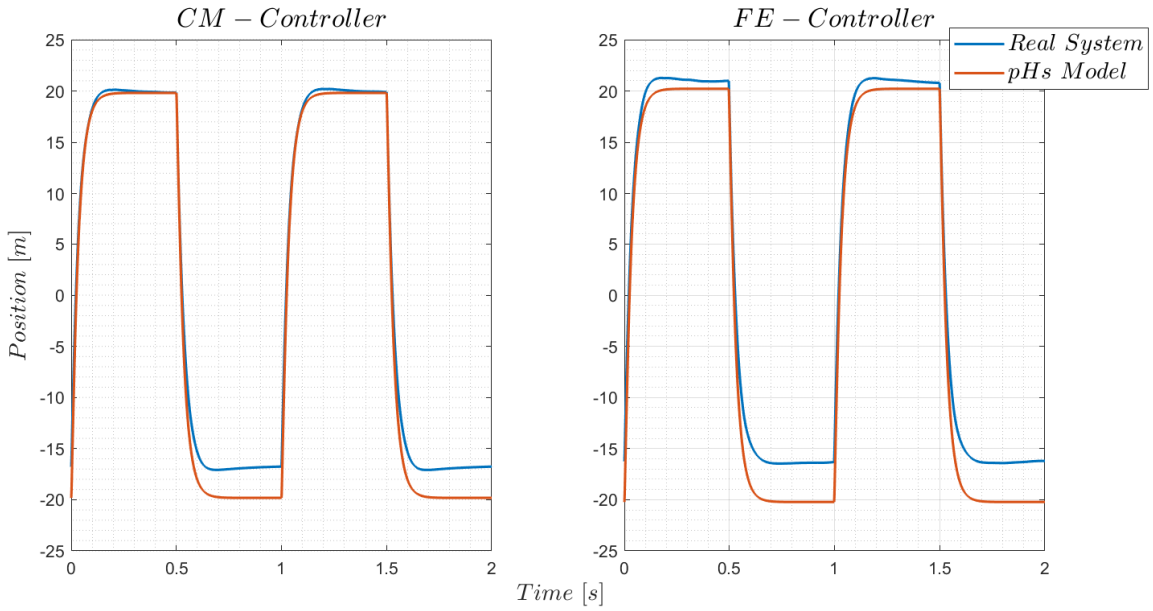


Figure 5.21: Test 2: Control action comparison between CM and Euler controllers

Chapter 6

Conclusions

In this Thesis, the continuous-time mathematical framework to analyze and control dynamical systems was presented. In particular, pHs were introduced and discussed, emphasizing the physical properties of this class of systems. Regarding controllers, two different control schemes were studied: one exploiting the passivity properties of pHs to design control laws based on their energetic characteristics, and the other using classical PI controllers designed through a linear control approach.

Discrete-time system theory was also discussed, and discrete-time approximations (DTA) for general systems were presented, with special attention to pHs and their controllers. Two DTA methods for pHs were introduced: the discrete-gradient method and the collocation method. On the controller side, DTA approximations were carried out using CM and TSS to resolve the causality issue of the plant states.

The piezoelectric phenomenon was analyzed, and PZAs were introduced. The hysteresis problem of these actuators was discussed, and a pHs model was proposed. Two position controllers for the PZA were developed: one using passivity-based control techniques, and the other based on a classical PI approach. Finally, both models and controllers were discretized using the methods presented in this work.

Using an experimental setup, the model parameters were obtained and validated. Using the updated parameters, the controller gains were adjusted, and the controller was implemented on the real system. Tests were conducted under varying sampling periods, and the performance was compared against that of a PI controller discretised using forward Euler.

As future work, it is proposed to develop controllers with integral action using Passivity-Based Control, and subsequently discretize them with the techniques presented in this Thesis. In addition, the aim is to implement the developed controllers in the same laboratory setup.

Bibliography

[1] H. K. Khalil, *Nonlinear systems*. Upper Saddle River, N.J.: Prentice Hall, 2002.

[2] W. M. Haddad and V. Chellaboina, *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. Princeton University Press, 2008.

[3] M. Salgado, J. Yuz, and R. Rojas, *Análisis de sistemas lineales*. Automática Robótica, Pearson Educación, 2005.

[4] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control System Design*. USA: Prentice Hall PTR, 1st ed., 2000.

[5] K. Ogata, *Modern Control Engineering*. USA: Prentice Hall PTR, 4th ed., 2001.

[6] R. Ortega, A. Van Der Schaft, I. Mareels, and B. Maschke, "Putting energy back in control," *IEEE Control Systems Magazine*, 2001.

[7] R. Ortega, A. van der Schaft, B. Maschke, and G. Escobar, "Interconnection and damping assignment passivity-based control of port-controlled hamiltonian systems," *Automatica*, 2002.

[8] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx, eds., *Modeling and Control of Complex Physical Systems: The Port-Hamiltonian Approach*. Berlin: Springer, 2009.

[9] A. van der Schaft, *L2-Gain and Passivity Techniques in Nonlinear Control*. Springer Publishing Company, Incorporated, 3rd ed., 2018.

[10] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration*, vol. 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second ed., 2006.

[11] K. J. Åström and B. Wittenmark, *Computer-controlled systems (3rd ed.)*. USA: Prentice-Hall, Inc., 1997.

[12] P. Albertos and S. Antonio, *Multivariable control systems: an engineering approach*. Springer Science & Business Media, 2006.

[13] E. F. Camacho and C. Bordons, eds., *Model predictive control*. Berlin Heidelberg: Springer-Verlag, 1999.

[14] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[15] P. Kotyczka and L. Lefèvre, "Discrete-time port-hamiltonian systems: A definition based on symplectic integration," *Systems Control Letters*, vol. 133, p. 104530, 2019.

[16] L. Gören-Sümer and Y. Yalçn, "Gradient based discrete-time modeling and control of hamiltonian systems," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 212–217, 2008. 17th IFAC World Congress.

[17] A. Macchelli, "Trajectory tracking for discrete-time port-hamiltonian systems," *IEEE Control Systems Letters*, vol. 6, pp. 3146–3151, 2022.

[18] A. Macchelli, "Control design for a class of discrete-time port-hamiltonian systems," *IEEE Transactions on Automatic Control*, 2023.

[19] M. Mogler, P. Kotyczka, and L. Lefèvre, "Discrete-time control by interconnection using energy-preserving collocation methods," *IFAC-PapersOnLine*, vol. 58, no. 6, pp. 172–177, 2024. 8th IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control LHMNC 2024.

[20] A. Aabid, M. A. Raheman, Y. E. Ibrahim, A. Anjum, M. Hrairi, B. Parveez, N. Parveen, and J. Mohammed Zayan, "A systematic review of piezoelectric materials and energy harvesters for industrial applications," *Sensors*, vol. 21, no. 12, 2021.

[21] D. Karnopp, "Computer models of hysteresis in mechanical and magnetic components," *Journal of the Franklin Institute*, vol. 316, no. 5, pp. 405–415, 1983.

[22] M. Al Janaideh, J. Mao, S. Rakheja, W. Xie, and C.-Y. Su, "Generalized prandtl-ishlinskii hysteresis model: Hysteresis modeling and its inverse for compensation in smart actuators," in *2008 47th IEEE Conference on Decision and Control*, pp. 5182–5187, 2008.

[23] J. Gan and X. Zhang, "A review of nonlinear hysteresis modeling and control of piezoelectric actuators," *AIP Advances*, vol. 9, p. 040702, 04 2019.

[24] M. Rakotondrabe, "Bouc–wen modeling and inverse multiplicative structure to compensate hysteresis nonlinearity in piezoelectric actuators," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 2, pp. 428–431, 2011.

[25] D. Habineza, M. Rakotondrabe, and Y. Le Gorrec, "Bouc–wen modeling and feedforward control of multivariable hysteresis in piezoelectric systems: Application to a 3-dof piezotube scanner," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1797–1806, 2015.

[26] J. Caballeria, H. Ramirez, and Y. L. Gorrec, "An irreversible port-hamiltonian model for a class of piezoelectric actuators," *IFAC-PapersOnLine*, vol. 54, no. 14, pp. 436–441, 2021. 3rd IFAC Conference on Modelling, Identification and Control of Nonlinear Systems MICNON 2021.

[27] I. Díaz, "A modular port based model and passivity based control approach for a class of piezoelectric actuators," 2023.

[28] A. v. d. Schaft and A. J. Schaft, *L2-Gain and Passivity in Nonlinear Control*. Berlin, Heidelberg: Springer-Verlag, 2nd ed., 1999.

[29] O. Gonzalez, "Time integration and discrete hamiltonian systems," in *Mechanics: From Theory to Computation*, (New York, NY), pp. 257–275, Springer New York, 2000.

[30] A. Harten, P. D. Lax, and B. V. Leer, "On upstream differencing and godunov-type schemes for hyperbolic conservation laws," *SIAM Review*, vol. 25, no. 1, pp. 35–61, 1983.

[31] A. Moreschini, S. Monaco, and D. Normand-Cyrot, "Gradient and hamiltonian dynamics under sampling," *IFAC-PapersOnLine*, vol. 52, pp. 472–477, 2019.

[32] M. Al Janaideh, J. Mao, S. Rakheja, W. Xie, and C.-Y. Su, "Generalized prandtl-ishlinskii hysteresis model: Hysteresis modeling and its inverse for compensation in smart actuators," in *2008 47th IEEE Conference on Decision and Control*, pp. 5182–5187, 2008.

[33] I. Díaz, H. Ramírez, Y. L. Gorrec, and Y. Wu, "Modular passivity-based modelling of piezoelectric actuators," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 31, no. 1, p. 2533294, 2025.

[34] N. Vignoli, *PI Control of a Piezoelectric Bending Micro-actuator for Precision Positioning Applications*. PhD thesis.