UNIVERSITÀ DEGLI STUDI DI CATANIA
Dipartimento di Ingegneria Elettrica Elettronica e Informatica

Dottorato di Ricerca in Ingegneria dei Sistemi, Energetica, Informatica e delle Telecomunicazioni- XXXIV ciclo
Ph.D. Thesis

New Results on Time-Delay Systems Towards Model-Based Design

Dott.ssa Ing. Loubna Belhamel

Tutor: Prof. Luigi Fortuna **Coordinator:** Prof. Paolo Pietro Arena

Academic Year 2020/2021

To my familiy.

ACKNOWLEDGEMENTS

First and foremost I am deeply indebted to my supervisor, Prof. Luigi Fortuna, for his guidance, supports and advices. This work is a result of his ideas, encouragement, and constructive criticism. He has helped me to mature and develop as a researcher and as a person. I am pleased to have work with him for all three years. I would also like to thank Prof. Maria Gabriella Xibilia for her good advices, cheerful mood, support whenever needed and the opportunities that she offered to me. I am also deeply thankful for all the suggestions and support from Prof. Arturo Buscarino. I would also like to thank Prof. Paolo Pietro Arena for all the opportunities that he have offered to me in my Ph.D. career.

Also deserving a special mention is the research group in STMicroelectronics, in particular Antonino Cucuccio, Nunzio Abate and my supervisor Gaetano Rascona who always trusted in my ability from the first day in ST.

To all my teachers, present and past, I am deeply indebted to you for the role you have played in taking me to where I stand today.

Last but not least, I would also like to thank all the members of my family, my mother Fatima, my father Abdulhak, my brother Mohammed and my sisters Amina, Ghizlan, and little Samia for their love, guidance and encouragement, which have played a defining role in my life. Many thanks to my husband, Leandro, for being the shining light in my life, and for his support and fortitude.

ABSTRACT

This thesis is devoted to the stability study of time-delay systems, a subject that has been vigorously pursued by such learned societies as diversely represented in mathematics, science, engineering, and economics. Time-delay systems, which are also sometimes known as hereditary systems, systems with memory, after effects and time-lag, represent a class of infinite-dimensional systems used to describe, among other types of systems, propagation and transport phenomena, population dynamics, economic systems, communication networks, and neural network models. The aim of the present thesis is to develop techniques and tools that may help to study the stability of commensurate time-delay systems. Stability analysis methods are developed based on the corresponding characteristic equation following a frequency sweeping test and constant matrix tests. Rewriting the quasi-polynomial equation, the coefficients can be found and then the roots of the characteristic equation can be plotted in a complex plane. If the roots cross the imaginary axis of the system, it is said to be Delay-Dependent Stable System (DDSS). A controller design procedure for Commensurate Multiple Time-Delay Systems (CMTDSs) is developed, able to transform the system into a Delay Independent Stable System (DISS). The controller based on a single parameter is used to make the system DISS. It may be determined by adopting different strategies, either analytical or graphical. Based on this theorem, a stability chart is partitioned into two regions, that are DDSS and DISS. As an application, it is demonstrated that model-based design can be used

to design systems with time delays. The stability analysis methods developed in this thesis are tailored and applied to find if the system is DDSS and to transform systems from DDSS to DISS.

Contents

List of Tables

List of Figures

LIST OF ABBREVIATIONS

R	The set of real numbers
R_+	The sets of positive reals
C	The set of complex numbers
C_+	The right half complex plane
C_-	The left half complex plane
D	The open unit disk
T	The unit circle
$\Re(\omega)$	The real parts of $\omega \in \mathbb{C}$
$\Im(\omega)$	the imaginary part of $\omega \in \mathbb{C}$
$\lambda(A)$	An enginvalue of the matrix A
$\rho(A)$	Spectrum radius of matrix A

Additional notations are defined in the text.

CHAPTER 1 INTRODUCTION

1.1 CHAPTER OVERVIEW

This chapter aims to provide state-of-the-art results and techniques adopted as a background for the development of the proposed theorems. The literature reviewed in this chapter is closely related to this thesis.

First, various problems arising due to delays in linear time-invariant (LTI) systems are discussed. Then, the basic principles of Time-Delay Systems (TDSs) including the transfer function, stability approaches, and some approximations for the general delay element are presented.

Finally, the basic notion of functional differential equations is reported, also including the definition of general TDSs form and some important proprieties of the TDSs characteristic equation are provided. Moreover, the principal research objectives and contributions are discussed along with chapters organizations.

1.2 INTRODUCTION TO TIME DELAYS SYSTEMS

TDSs are also called systems with after effect or dead-time, hereditary systems, equations with deviating argument, or differential-difference equations. They belong to the class of functional differential equations (FDEs) which are infinite-dimensional, as opposed to ordinary differential equations (ODEs).

There exists a great number of monographs devoted to this field of active research (at least 40 English-language books since 1963). In this section, several TDSs fields are presented. This should give the reader a glimpse of how widely time delays may occur in practice.

TDSs are frequently encountered in various fields, such as economy [1], chemistry [2], network control systems [3] and control engineering systems [4]. A time-delay term is also used to model several biological processes [5–7].

For example, in economics, delays can arise in data analysis or strategic decisions. As shown in [8], the analysis of the time delay of information flow may help to better understand stock market processes and control its behavior at times of drastic instabilities.

Delays arise also in biology. For instance, a population can grow only after the offspring mature and become reproductive, Models of reaction chains or a transport process, gestation times, incubation periods, transport delays, or lump biological processes take into account delay terms. In [9], the delays can be presents also in Epidemic Models such as COVID-19 [10], MERS-CoV and SARS-CoV [11] swine flu viruses H1N12 [12] and H5N1 which have sparked a deadly outbreak in some countries and spread into other parts of the world.

In network control systems [13], delays can be found in the communication network. Usually, in this case two types of delays can occur: internal delays(due to specific internal dynamics of a given node) and external delays (related to the communication process, due to the information transmission and processing).

In control engineering, delay terms arise from mass transport phenomena in stirred-tank reactors and flow temperature-composition control [14,15]. Moreover, many systems in control engineering are based on the existence of a time-delay, such as radars [16] and sonars [17] whose working principle relies on the time-delay between the emitted wave and the reception of the echo.

Indeed, the presence of a human operator in a control loop is modeled as a time-delay, an aspect of crucial importance in the theory of human-machine interaction [18].

Besides, in industrial automation time delays are a common feature, always appearing in plants and systems involved in automatic factories. In the evaluation of just-in-time production, the queue delay systems are the key points of the optimization. Time delays are present in the temperature control system for our showers [19] and, particularly referring to the internal combustion engines [20], also the production of torque [21] is delayed with respect to the required one.

An interesting series of examples of systems with time delays include different classes of systems:

- fluid flow models for a congested router in TCP/AQM controlled network [22];
- car following systems [23];
- heating systems [24].

In the last decade, particular interest has been also devoted to internet congestion [25] with many contributions proposed to understand the role of time-delay in this field. In particular, the topic of robust control of time-delay systems is dealt with in several books [19, 26].

Important applications in underwater control systems [27], in biosystems analysis and control, and the area of mathematical modeling have been also discussed. Delays terms also model circulation dynamics of hormones in the bloodstream [28], and the dynamics of chronic myelogenous leukemia [29].

Delays also found in teleoperation [30] telesurgery [31], the coordination of unmanned vehicles [32, 33], decentralized and collaborative control of multiple agents [34] , [35], synchronization and haptics [36].

The presence of time delays may be either beneficial or detrimental to a dynamic system. A feedback system that is unstable without delays can become stable when a delay is added [37], and, on the contrary, a system that is stable without delays may become unstable for some delays [38, 39]. That is why the topic receives substantial attention from the control research community [40, 41].

For example, appropriate adjustment of the spindle speed helps in tuning the delay to avoid chattering in metal machining, while intentionally adding delays to decision-making allows supply-chain managers to observe consumer trends to make better purchasing and stocking decisions [42].

1.3 BASIC PRINCIPLES OF TIME-DELAY SYSTEMS

Let us consider the continuous-time linear system S_1, where the output $y(t)$ is related to the input $u(t)$ by the relationship $y(t) = u(t-\tau)$. This system represents an ideal time-delay. In fact, the output is a replica of the input after a delay τ. The transfer function of this system is given by:

$$G(s) = e^{-s\tau} \tag{1.1}$$

which is a direct application of one of the properties of the Laplace transform.

The characteristic of $G(s)$ is that it is an all-pass stable system, i.e.,$|G(j\omega)| = 1, \forall\omega$. The system is BIBO stable, the unique singularity is for $s \to -\infty$ and the unique zero is at $s \to \infty$. The system belongs to the class of infinite dimensional linear systems.

Let us now consider the closed-loop system reported in Figure. 1.1. The transfer function is

$$F(s) = \frac{ke^{-s\tau}}{1 + ke^{-s\tau}} \tag{1.2}$$

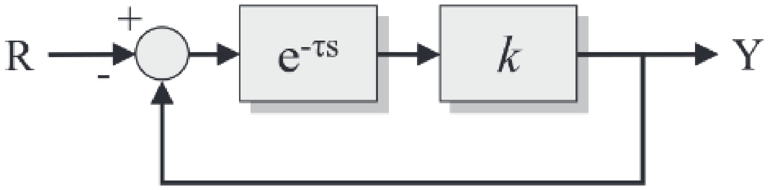


Figure 1.1: A closed-loop system with a time-delay block.

In order to check its stability the poles of the systems must be computed by solving

$$1 + ke^{-s\tau} = 0 \tag{1.3}$$

Equation (1.3) is a transcendental equation that can be rewritten as

$$ke^{-\sigma\tau}e^{-j\omega\tau} = -1 \tag{1.4}$$

that is a complex equation corresponding to the following conditions on the real and imaginary part:

$$\begin{gathered} ke^{-\sigma\tau}\cos\omega\tau = -1 \\ ke^{-\sigma\tau}\sin\omega\tau = 0 \end{gathered} \tag{1.5}$$

The second condition yields $\sin\omega\tau = 0$, that is solved by $\omega\tau = i\pi$, with $i = 1,2,...,n$. The first condition implies that the index i must be odd. In this case, the solution is given by $\sigma = \frac{\ln k}{\tau}$. The system is therefore stable if $0 < k < 1$.

The same result can be obtained considering the small-gain theorem. In fact, since it is an unitary control feedback scheme, then $|ke^{-j\omega\tau}|$ must be less than 1, $\forall\omega\tau$ that leads to the condition $0 < k < 1$.

Another possibility to derive the same result is to consider the Nyquist plot and determine the number of encirclements of the critical point $(-1,0)$. In fact, in this case, for each $k > 1$ the critical point is encircled and thus the system is unstable.

From this simple example, the poles of time-delay system are an infinite number. In order to have information about the stability of a time-delay system, several approaches can be used:

1. the analytical one, that means to find the infinite roots of the characteristic equation;

2. the classical approach in the frequency domain by using the Nyquist criterion and the Bode diagrams.

The stability of delayed systems depends not only on the static gain k, but also on the time-delay τ. The next example illustrates another important case study.

Ex 1.3.0.1 Let us consider the classical control scheme shown in Figure. 1.2. It can be physically interpreted as a delay speed control system for a motion controlled system. The controlled system may represent a car, an airplane, a bicycle and so on, whereas the control action is performed with some delay, due to the physiological characteristics of the human response.

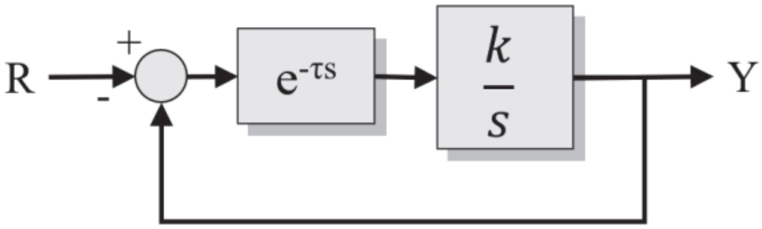


Figure 1.2: The closed-loop system used in Example 1.3.

To obtain the stability condition, the phase margin of the open-loop system must be considered. Stability requires that it is positive. In order to compute it, the so-called crossover frequency can be calculated as

$$|\frac{k}{j\omega_c}e^{-j\omega_c\tau}| = 1 \tag{1.6}$$

therefore, $\omega_c = k$. The phase margin is given as follows:

$$m_\phi = \pi - \psi_{\omega_c} = \pi - \frac{\pi}{2} - \omega_c\tau = \frac{\pi}{2} - \omega_c\tau = \frac{\pi}{2} - k\tau \qquad (1.7)$$

Then, the stability condition yields $k\tau < \frac{\pi}{2}$.

It is evident that high values of the time-delay τ require to decrease the gain, at the expenses of the precision of the control. On the contrary, for small values of the time-delay τ, an higher gain can be used, with better performance.

The examples reported above show the effects of the time-delay in simple case studies. More in general, systems may include more than a single delay. In similar cases, in order to get some insights on the system behavior, it is often convenient to introduce an approximation of the various time-delay elements. It therefore useful to get a rational function approximation of the general delay element $e^{-s\tau}$ that reflects its important properties.

The Padé approximation is the one more commonly used in many applications. It is derived by matching the first coefficient of the Taylor expansion of $e^{-s\tau}$ with that of a rational transfer function of order n that must have all-pass characteristics. Therefore, if the approximation function is indicated as $R_n(s) = \frac{N_n(s)}{D_n(s)}$, we have

$$D_n(s) = N_n(-s) \qquad (1.8)$$

1.4 FUNCTIONAL DIFFERENTIAL EQUATIONS

Functional differential equations can be used to describe time-delay systems. To formally introduce the concept of functional differential equations, let $C([a,b],\Re^n)$ be the set of continuous functions mapping the interval $[a,b]$ to $\Re^n$. In many situations, one way wish to identify a *maximum time-delay r* of a system. In this case, it is important to set the continuous functions mapping $[-r,0]$ to $\Re^n$, whose notation can be simplified with $\mathscr{C} = \mathscr{C}([-r,0],\Re^n)$. For any $A > 0$ and any continuous function of time $\psi \in \mathscr{C}([t_0 - r, t_0 + A],\Re^n)$, and $t_0 \leq t \leq t_0 + A$, let $\psi_t(t+\theta)$, $-r \leq \theta \leq 0$. The general form of a *retarded functional differential equation (RFDE)* (or functional differential equation of *retarded type*) is

$$\dot{x}(t) = f(t, x_t) \tag{1.9}$$

where $x(t) \in \Re^n$ and $f : \mathscr{R} x \mathscr{C} \longrightarrow \mathscr{R}^n$. Equation (1.9) indicates that the derivative of the state variables x at time t depends on t and $x(\xi)$ for $t - r \leq \xi \leq t$. As such, to determine the future evolution of the state, it is necessary to specify the initial state variables $x(t)$ in a time interval of length r, say, from $t_0 - r$ to t_0, *i.e.,*

$$\dot{x}_{t_0} = \phi, \tag{1.10}$$

where $\phi \in \mathfrak{C}$ is given. In other words, $x(t_0 + \theta) = \phi(\theta), -r \leq \theta \leq 0..$

For examples of retarded functional differential equations, let us consider

$$\dot{x}_{t_0} = ax(t) + b(x-r) + \cos(\omega t), \tag{1.11}$$

$$\dot{x} = (2 + \sin\omega t)x(t), \tag{1.12}$$

$$\dot{x}_{t_0} = \phi, \tag{1.13}$$

A retarded functional differential equation may also involve higher order derivatives, which is known as a higher order *RFDE*.

1.5 CHARACTERISTIC ROOTS

The state-space representation of a linear time-invariant (LTI) system with time delays is the following:

$$\dot{x}(t) = A_0 x(t) + \sum_{k=1}^{m} A_k x(t-\tau_k) \tag{1.14}$$

where $A_0, A_k \in \mathbb{R}^{nxn}$ with $k = 0,...,q$ are constant state matrices and $\tau_i \geq 0$ for $i = 1,\ldots,q$ are time delays. Two types of TDSs can be distinguished, as reported below.

Definition 1 *Incommensurate Time-Delay System*
A TDS (1.15) is said uncommensurate if r_k (k=1,...,m where m is the number of delays) are free parameters, i.e. the delays are rationally independent numbers.

$$\dot{x}(t) = A_0 x(t) + \sum_{k=1}^{m} A_k x(t-r_k) \tag{1.15}$$

where A_0, $A_k \in R^{n\times n}$ are constant system matrices, $r_k \geq 0$, are time delays and x(t) $\in R^{n\times n}$ is the state vector.

Definition 2 *Commensurate Time-Delay System*

A TDS (1.16) is said commensurate if r_k (k=1,...,m) are rationally dependent numbers, i.e. r_k = $k\tau$.

$$\dot{x}(t) = A_0x(t) + \sum_{k=1}^{m} A_kx(t - k\tau) \tag{1.16}$$

In this thesis, only the commensurate TDSs are taken into account.

The stability of those systems is fully determined by its characteristic quasi-polynomial $a(s,\tau_k)$, which is given by

$$a(s,\tau_k) = det(sI - A_0 - \sum_{k=1}^{q} A_ke^{-s\tau_k}) \tag{1.17}$$

or equivalently:

$$a(s,e^{-\tau s}) = \sum_{k=0}^{q} a_k(s)e^{-k\tau s}, \tag{1.18}$$

where

$$a_0(s) = s^n + \sum_{i=0}^{n-1} a_0is^i, \quad a_k(s) = \sum_{i=0}^{n-1} a_kis^i, k = 1,\dots,q. \tag{1.19}$$

Definition 3 *The characteristic quasi-polynomial is said to be stable if all its roots $a(s,\tau)$ lie in the open left half plane. It is said to be delay-independent stable if this condition is valid for all $\tau \geq 0$. System is stable if and only if its characteristic quasi-polynomial is stable.*

It is convenient to introduce the variable $z = e^{-\tau s}$, and write (1.18) as a bi-variate polynomial

$$a(s,z) = \sum_{k=0}^{q} a_k(s) z^{-k}, z = e^{-\tau s} \tag{1.20}$$

Notice that the order of $a_0(s)$, often known as the ”principal term”, is higher than the order of any $a_k(s)$, $k = 1,2,\ldots,q$. Throughout this thesis, it is assumed that the system (1.16) is stable for $\tau = 0$ or, equivalently, $a(s,1)$ is stable. Following the above continuity argument, the smallest deviation of τ from $\tau = 0$ such that the system becomes unstable can be determined as

$$\bar{\tau} := \min\{\tau \geq 0 |\ a(j\omega, e^{-j\tau\omega}) = 0 \quad \text{for some} \quad \omega \in \mathbb{R}\}. \tag{1.21}$$

$\bar{\tau}$ is called the *delaymargin* of the system. For any $\tau \in [0,\bar{\tau})$ the system is stable and, whenever $\bar{\tau} = \infty$, the system is stable independent of delay. Note that for any finite $\bar{\tau}$, the frequency at which $a(j\omega, e^{-j\bar{\tau}\omega}) = 0$ represents the first contact or crossing of the characteristic roots from the stable region to the unstable one. Note also that multiple crossings may exist. However, since only finitely many unstable roots may be in the right half plane, there are only a finite number of zero crossings.

Moreover, since $a(s, e^{-\tau s})$ is a real quasi-polynomial, all its complex roots appear in complex conjugate pairs; that is, it satisfies the *conjugate symmetry* property. Consequently, it suffices to consider only the zero crossings at positive frequencies. Let

$a(j\omega_i, e^{-j\theta_i}) = 0, \omega_i > 0, \omega_i \in [0, 2\pi], i = 1, 2, \ldots, N.$

Furthermore, define $\eta_i = \theta_i/\omega_i$. It is clear that

$$\bar{\tau} := \min_{1 \le i \le N} \eta_i = \min\left\{\frac{\theta_i}{\omega_i} | \omega_i > 0\right\}. \tag{1.22}$$

This gives a general formula for computing the delay margin. More generally, it can be assumed, with no loss of generality, that $\eta_1 < \eta_2 < \cdots < \eta_N$. It follows that the system is stable for all $\tau \in (\eta_i, \eta_{i+1})$ whenever it is stable for some $\tau^* \in (\eta_i, \eta_{i+1})$. Then, this allows to ascertain a system's stability in the full range of delay values, beyond the interval determined by $\bar{\tau}$. It also indicates that the stability at $\tau = 0$ can be made without loss of generality.

By using the bi-variate polynomial representation (1.20), it is rather evident that can be found by solving the imaginary roots $s \in \partial\mathbb{C}_+$ and the unitary roots $z \in \partial\mathbb{D}$ of $a(s,z)$, giving rise to a stability criterion commonly referred to as *two-variable criterion.*

The two-variable criterion appears to be the origin of many classical stability tests for systems with commensurate delays, which attempt to solve the bi-variate polynomial (1.20) in one way or another.

In particular, most of the classical tests attempt to accomplish this by means of eliminating one variable, thus converting the stability problem to one free of delay, then seeking the solution of polynomials of one single variable.

In the next chapter, a number of tests related to the previously mentioned theory are presented. These sample tests are rather representative of classical results and should give the reader the essential flavor of the two-variable criterion.

1.6 STATE-OF-THE-ART GAPS

As mentioned above, several works have been published on CTDSs subject. However, there are cases in which some gaps can be overcome, which are the objectives of this thesis. These gaps are summarized below:

- In order to find the quasi-polynomial coefficients, the literature based on the symbolic method is hard to compute when the multiple delays are present within TDSs;
- when studying the stability of CTDSs, there is a list of analytical and graphical methods that allows finding the roots of the characteristic equation and the delay margin. However, the approaches have a list of drawbacks. In case of analytical method, which includes:

  1. Hermite Matric Formation [65,66]
  2. Elimination of Transcendental terms [67]
  3. Matrix Pencil [68]
  4. Rekasius Substitution [70]
  5. Kronecker Multiplication [69]
  6. Nyquist criteria path immaginary axis and a half circle [71–76]

  they all contain some small numerical errors, high degree complex polynomial to be solved and some of them require special attention avoiding the false solutions besides hardness in the numerical algorithm.

Instead, for the graphical methods including the one based on root-loci, it takes several steps to obtain the stability results;

- many theoretical results reported in literature provide necessary and sufficient conditions for the TDSs stability, but most of them deal with systems with only one or two delay parameters [45–48]. In [49] some conditions for ensuring a DDSS were proposed, based on finding a time-delay range for which the system is stable. In [26], a method based on the analytic curve perspective is introduced to calculate the spectrum of a time-delay system. It is shown that the asymptotic behavior of the critical imaginary roots can be investigated through the Puiseux series. A recent result was presented in [52], which shows that the asymptotic behavior of imaginary roots with multiplicity can be completely characterized by the Newton-Puiseux series. However, all the approaches presented in the literature are based on finding the stable regions in the delay parameter space using the concept that the crossing frequencies are only dependent on the delay. On the other side, some results were discussed in the literature which gives necessary and sufficient conditions for DISSs [53, 54], that require the computation of constant matrices in the frequency domain. However, although this approach allows to know whether the system is DDSS or not, it does not provide a strategy to render a DDSS into a DISS;
- the main idea behind the MBD is the development of a complete

system in a virtual environment, to reproduce the expected behavior of the real system and to predict its performances before the building. The MBD can be applied to model different systems such as embedded systems [77, 78], on-board power systems [79], protection devices [80]. However, a dedicated modeling approach for a time-delay systems does not exist in the MBD literature.

1.7 RESEARCH OBJECTIVES

The contents of this thesis are centered on the theme of stability of commensurate time-delay systems and their application. The contribution of this work is to overcome the State-Of-The-Art drawbacks previously mentioned. The main objectives are:

- To develop a simplified and efficient method to find the quasi-polynomial coefficients of CTDSs especially for systems with multiple delays, in order to overcome the hard computation of the symbolic approach;
- to design a graphical method for stability delay-dependent analysis, thus allowing to immediately obtain the imaginary roots in the complex plane and the correspondent delay margin, that avoids inaccuracy and complex computations;
- to present a controller design procedure able to transform the system into a delay-independent stable system without any condition of the delay parameter as in the literature;

- to draw a stability chart for delay independent stability that depends only on two parameters;
- to model an application example on how time-delay affects the system stability towards a model-based design approach.

1.8 CHAPTERS ORGANIZATION

This chapter has begun with several areas where time delays play an important role. Then, it continues to provide an introductory exposition of some basic concepts and results for stability analysis, such as functional differential equation representation and characteristic quasi-polynomial. Moreover, a state-of-the-art, particularly regarding the CTDSs stability is reported. Finally, a brief outline of research objectives is shown.

Chapter 2 focuses on systems with commensurate delays only. It shows the correspondent results presented in the literature which helps to build the main contributions of this thesis. It begin by presenting the main approaches to find the imaginary characteristic equation roots and then it shows both frequency sweeping and constant matrix tests. They are necessary and sufficient conditions for delay-dependent and delay-independent stability, and both of them require computing matrix pencils.

Chapter 3 is divided into two main sections which regard the two first contributions of this work. The former is developing a numerical procedure to obtain the pseudo-polynomial characteristic equation coefficients. The method is formulated in terms of an interpolation problem, and it is based on the generation of a suitable set of random numbers to find the characteristic equation coefficients. The latter is based on the development of a graphical approach that provides the necessary and sufficient condition for stability dependent on delay. Finally, the chapter provides a comparison between one of the analytical approaches presented in Chapter 2 and the graphical approach in terms of complexity, degree of the polynomial, and numerical errors.

Independent delay stability systems approaches are addressed in Chapter 4 and it also contains other two contributions of this thesis. Firstly, it introduces a single controller gain that allows making the system DISS. The approach could be applied to any CTDS, independently from the number of delays. It is based on two fundamental steps: the analysis of the system stability (both delay-dependent or independent) and, in case of a DDSS, the design of the gain able to transform it in a DISS. A control gain parameter, that allows shifting the roots from the Right Half Complex Plane (RHCP) to the Left Half Complex Plane (LHCP) for any values of delay, is proven. Finally, visualization of asymptotic stability in the form of stability chart is demonstrated using the controller gain parameter and another variable.

Chapter 5 contains the last contribution, that concentrates on Model-Based Design approach with time-delay. It shows how the model-based design approach can be used to implement the time delay in a feedback control scheme. The block time delay in *Simulink* is adopted to approximate the time delay. An application based on STM32 Motor Control

Embedded Software System will be reported to show how time delays affect such systems.

CHAPTER 2 STABILITY OF SYSTEMS WITH COMMENSURATE DELAYS - STATE-OF-THE-ART

2.1 CHAPTER OVERVIEW

This chapter presents how the stability of systems with commensurate delays has been exploited in the literature. From Chapter 1, it is clear that the stability of an LTI delay system can be completely characterized by its characteristic roots, i.e., the solutions to its characteristic equation.

Two stability notions, known as delay-independent stability (or stability independent of delay) and delay-dependent stability (or stability dependent of delay) are covered in this chapter.

The chapter opens with discussing a number of classical tests to study the system stability using two-variable criterion. Then, it gives a general method to determine if the system is Delay Dependent Stability (DDS) or Delay Independent Stability (DIS). It also includes the Schr-Cohn method, Elimination of transcendental terms, Matrix pencil and Rekasius substitution. These methods allow to find the corresponding frequencies of imaginary characteristic roots if they exist. Then, the main results related to DISs conditions for LTI systems with commensurate delays will be discussed.

In particular, for systems with commensurate delays, a frequency sweeping test is presented which requires computing the spectral radius of a frequency-dependent matrix.

Finally, it covers the visualization of asymptotic stability in the form of stability charts.

2.2 CLASSICAL STABILITY TESTS

2.2.1 2-D Stability Tests

Given that the characteristic quasi-polynomial can be presented as a bi-variate polynomial, therefore, it can be treated as the characteristic polynomial of a *2-D* system, and its stability would be analyzed as in the case of a *2-D* polynomial. Indeed, let consider the bi-linear transformation

$$s = \frac{1+\lambda}{a-\lambda} \tag{2.23}$$

which maps s from the open right half complex plane C_+ to λ in the open unit disk D. Construct the $2-D$ polynomial

$$b(\lambda,z) := (1-\lambda)^n a(\frac{1+\lambda}{1-\lambda},z) \tag{2.24}$$

It is evident that $a(s,z) = 0$ for some $(s,z) \in \partial Dx\partial D$. In addition, the quasi-polynomial $a(s,e^{-\tau s})$ has no root in $\bar{C}_+$ if and only if $b(\lambda,z)$ is stable; stability of a *2-D* polynomial means that all its roots lie outside the closed region $\bar{D}x\bar{D}$. Hence, to verify if the system is stable independent of delay, it suffices to check whether the *2-D* polynomial $b(\lambda,z)$ is stable. Moreover, to determine whether the stability is delay-dependent, it is necessary to calculate the roots of $b(\lambda,z)$. Stability of *2-D* polynomials and *2-D* systems were deeply covered in the theory of signal processing. The equivalence noted here between the stability of a time-delay system and that of a *2-D* polynomial enables us to draw upon the existing analysis techniques developed extensively for the latter, although *2-D* stability tests themselves generally pose a rather formidable computational task.

In broader terms, one may view the bi-variate polynomial $a(s,z)$ as a *2-D* polynomial and tackle the stability problem directly. Define the conjugate polynomial

$$\bar{a}(s,z) := z^q a(-s, z^-1). \tag{2.25}$$

By the conjugate symmetry of $a(s,z)$, it follows that $(s,z) \in \partial C_+ x \partial D$ is a root of $a(s,z)$ if and only if it is also a root of $\bar{a}(s,z)$. Thus, in order to find the roots of $a(s,z)$ on $\partial C_+ x \partial D$, it suffices to solve the simultaneous polynomial equations

$$a(s,z) = 0, \tag{2.26}$$

$$\bar{a}(s,z) = 0. \tag{2.27}$$

When no solution exists, and when the system is stable in the delay-free case, it must also be stable independent of delay. Otherwise, when the two equations do admit a common solution, it is possible to eliminate one variable, resulting in a polynomial of one single variable. For example, we may eliminate s and obtain a polynomial in z, i.e. $b(z)$. If $b(z)$ has no unitary root, we may again conclude that the system is stable independent of delay. Otherwise, we may proceed to find all the unitary roots z_i of $b(z)$.

There are only a finite number of such roots since $b(z)$ is a polynomial. For each z_i, $a(s,z_i)$ is a polynomial of the variable s, which admits only a finite number of possible roots $s_i \in \partial C_+$. Thus, all the roots $(s_i,z_i) \in \partial C_+ x \partial D$ such that $a(s_i,z_i) = 0$ can be found. Since the bi-variate polynomial satisfies the conjugate symmetry property, only the s_i on the positive imaginary axis need to be considered, where $s_i = j\omega_i$, $z_i = e^{-\omega i}$, and, $\omega_i > 0$, $\omega_i \in [0,2\pi]$. The delay margin can then be determined using (1.22).

2.2.2 Direct Method

This section is devoted to illustrate the *directmethod* that provides a criterion to evaluate the stability of quasi-polynomials of the form:

$$p(s,\tau) = p_0(s) + p_1(s)e^{-s\tau} \tag{2.28}$$

under the hypothesis that the system without delay is stable, that means the roots of $p(s,0)$ are all in the left-half plane. The limit value of the delay, τ^*, for which the delay system is stable, is found by considering when the characteristic equation $p(s,\tau)$ has some solution on the imaginary axis. Thanks to the complex conjugate symmetry of the complex roots, this root is also a solution of the equation $p(-s,\tau) = 0$ for the same value τ. Thus, simultaneous solutions of $p(s,\tau) = 0$ and $p(-s,\tau) = 0$ for $s = j\omega$ allow to obtain the following equations:

$$\begin{aligned} p_0(j\omega) + p_1(j\omega)e^{-j\omega\tau} &= 0 \\ p_0(-j\omega) + p_1(-j\omega)e^{j\omega\tau} &= 0 \end{aligned} \tag{2.29}$$

that prompts for the derivation of the exponential term as follows:

$$e^{j\omega\tau} = -\frac{p_0(-j\omega)}{p_1(-j\omega)} \tag{2.30}$$

Substituting this expression in 2.29 it results

$$p_0(j\omega)p_0(-j\omega) - p_1(j\omega)p_1(-j\omega) = 0 \tag{2.31}$$

or equivalently

$$|p_1(j\omega)|^2 - |p_0(j\omega)|^2 = 0 \tag{2.32}$$

When equation (2.31) (or equation (2.32)) admits a solution, i.e. , then, the limit delay for stability can be derived from (2.30) with $= \bar{\omega}$. Otherwise, if equation (2.31) (or equation (2.32)) does not have any solutions, then the system with characteristic equation $p(s,\tau)$ is stable for any value of τ.

2.3 DELAY DEPENDENT STABILITY

To determine if the system stability is delay dependent or independent, it must be verified if the roots of the characteristic equation cross the imaginary axe in the complex plane. In case of delay dependent systems stability, the transition points from stable to unstable behavior are given by the pure imaginary roots.

There exist five frequency approaches in the literature to find the imaginary roots of the characteristic equation:

- Schur-Cohn method
- Elimination of transcendental terms
- Matrix pencil, Kronecker sum method
- Kronecker multiplication and elementary transformation
- Rekasius substitution

The first two and the last approaches are described in details below.

2.3.1 Schur-Cohn Method

To obtain the *Schur-Cohn* matrix, the first step is rewriting (1.20) by multiplying it with $z^k, for\, k = 0, 1, \ldots, q-1$. This generates q equations in terms of z^k, $k = -q, \ldots, -1, 0, 1, \ldots, q-2, q-1$, which are $2q$ linearly independent terms. Let us consider the companion equation, $a(-s, z) = 0$, which also satisfies for $s = j\omega$ due to the fact that the imaginary characteristic root $j\omega$ always appears as a complex conjugate pair.

$$\overline{a}(s,z) = a(-s,z) = \sum_{k=0}^{q} \overline{a}_k(s) z^k = \sum_{k=0}^{q} a_k(-s) z^k \tag{2.33}$$

Then, multiplying $\overline{a}(s,z)$ with z^{-k}, $k = 1, 2, \ldots, q$, it generates other q equations in terms of the same *2q* linearly independent terms z^k. Both of these sets of q equations can be combined in a single matrix equation as: $\Sigma(s)E_2 = 0$

$$E_2 := \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ \vdots \\ e_q \\ e_{-q} \\ e_{-q+1} \\ \vdots \\ \vdots \\ e_{-1} \end{bmatrix} \tag{2.34}$$

where e_k represents z^k as a shorthand notation and $\sum(s)$ is known to be as Schur-Cohn matrix:

$$\Sigma(s) := \begin{bmatrix} \Sigma_1(s) & \Sigma_2(s) \\ \Sigma_2{}^H(s) & \Sigma_1{}^H(s) \end{bmatrix} \tag{2.35}$$

where $\Sigma_1{}^H$ implies the hermitian of Σ_1. Σ_1 and Σ_2 are:

$$\Sigma_1(s) = \begin{bmatrix} a_0(s) & 0 & \dots & 0 \\ a_1(s) & a_0(s) & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ a_{q-1}(s) & a_{q-2}(s) & \dots & a_0(s) \end{bmatrix} \tag{2.36}$$

$$\Sigma_2(s) = \begin{bmatrix} a_q(s) & a_{q-1}(s) & \dots & a_1(s) \\ 0 & a_q(s) & \dots & a_2(s) \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & a_q(s) \end{bmatrix} \tag{2.37}$$

The determinant of the matrix (2.35) for $s = j\omega$ is:

$$B(s) = \det\Sigma(s) = (-1)^q \mid a_q(s) \mid^{2q} \prod_{i,j=1}^{n} (1 - z_i\bar{z}_j), z = e^{\tau s} \tag{2.38}$$

where z_i, $i = 1,\ldots,q$, are the roots of the polynomial $a(z)$ for fixed s. Therefore, for all frequencies $\omega > 0$, there is a $z \in \partial\mathbb{D}$ ($\mathbb{D}$ is the open unit disk) such that $a(s) = a(z) = 0$ whenever $B(s) = 0$.

By solving the equation $B(s) = 0$, all such $\omega > 0$ where $a(z) = a(s,z)$ has a root on the unit circle $\partial\mathbb{D}$ can be found. Since $B(s)$ defines a polynomial in ω, the solutions can be found by solving the eigenvalues of a constant matrix. Obviously there are only a finite number of solutions.

2.3.2 Elimination of Transcendental Terms

This procedure follows the similar starting premise as in *Schur-Cohn* methodology. If the characteristic quasi-polynomial $a(s,\tau)$ have an imaginary roots the corresponding $\bar{a}(s,\tau)$ should also have the same roots. By multiplying it with $e^{-k\tau s}$ leads to:

$$e^{-k\tau s}\bar{a}(s,\tau) = \sum_{i=1}^{k} \bar{a}_i(s)e^{(i-k)\tau s} = 0 \tag{2.39}$$

Then, it can be eliminated the highest commensurate term (i.e.$e^{-p\tau s}$) between equation (1.20) and equation (4.102) yielding a new equation:

$$a_1(s,\tau) = \sum_{i=1}^{k-1} a_i(s)^{(1)} e^{i\tau s} = 0 \tag{2.40}$$

which has degree of *k-1* . If this procedure of eliminating the highest degree terms k is repeated successively, until it arrives at

$$a_k(s,\tau) = a_0^{(k)}(s) = 0 \tag{2.41}$$

an algebraic characteristic equation with no transcendentally left. It can be shown that a_0^k is a polynomial of degree $n2^k$. Notice that due to the successive substitution of ”s” with ”$-s$” during the manipulations, the imaginary roots of the original characteristic equation a are preserved, although the degree of the s terms in polynomials $a_i(s)$ continuously increases. Ultimately there remains only $n2^k$ finite roots of $a_k(s)$ instead of the infinitely many roots of the original $a(s,\tau)$. It is guaranteed that only the imaginary roots of these two equations are identical. Therefore searching for the imaginary roots of $a_k(s)$ is the sufficient procedure for the mission.

2.3.3 Rekasius Substitution

Rekasius substitution is based on a substitution in equation (1.20).

$$e^{-p\tau s} = \frac{1-\tau s}{1+\tau s}, \quad \text{when} \quad s = \omega i \quad \text{only}, \tau \in \Re \quad \text{where} \tag{2.42}$$

$\tau = \frac{2}{\omega}[tan^{-1}(\omega\tau) \pm l\pi], l = 0, 1, \dots$

This exact substitution creates a new characteristic equation

$$a(s,\tau) = \sum_{k=0}^{p} \bar{a}_k(s)\left(\frac{1-\tau s}{1+\tau s}\right)^k = 0 \qquad (2.43)$$

Multiplying (2.43)with $1 + \tau s^p$ leads to:

$$\sum_{k=0}^{p} \bar{a}_k(s)(1+\tau s)^{p-k}(1-\tau s)^k = 0 \qquad (2.44)$$

Considering that $a_k(s)$ are ordinary polynomials, equation (2.44) is nothing other than a polynomial in s with parameters coefficients in τ. Since the system in equation (1.16) is retarded type, the highest degree term of s is n and it is in $a_0(s)$. Therefore, equation (2.44) is a polynomial of s in degree n + p. The question is to determine all $\tau \in \mathbb{R}$ values, which cause imaginary roots of $s = \omega i$. This can be achieved by forming the Routh's array of the equation (2.44), and setting the only term in the 1s row to zero. It can be shown that this polynomial is of degree np in τ, of which only the real roots are searched. Once these roots are determined the corresponding crossing frequencies $s = \omega i$ can be found using the auxiliary equation, which is formed by the $s^2 i$ row of the Routh's array.

Notice that, the $s^2 i$ row has two terms, which are functions of τ. They must agree in sign for those τ values to yield imaginary roots. Final results are exhaustive in detecting all the imaginary characteristic roots we set out to solve. In the case of degenerate imaginary roots at the origin, $s = \omega i$ with ωi, one needs to check in addition, the constant term in equation (2.44) with no s term; if

$$\sum_{k=0}^{p} a_k(0) = 0 \tag{2.45}$$

is satisfied or not. If it does there is at least one root at $s = 0$, which remains there for all $\tau \in \mathbb{R}^+$. It is easy to determine if this root is a multiple root for some τ values.

2.4 DELAY INDEPENDENT STABILITY

Frequency domain delay-independent and necessary and sufficient stability conditions will be presented.

2.4.1 Frequency Sweeping Method

The characteristic equation (1.17) is said to be DISS if and only if it occurs

1. $a_0(s)$ is Hurwitz

2. $a_0(s) + \sum_{k=1}^{q} a_k(s)$ is Hurwitz

3. $\rho(\mathrm{M}(j\omega)) < 1, \qquad \forall \omega > 0,$

where $\rho(\mathrm{M}(j\omega))$ indicates the spectral radius of the matrix M defined as

$$\mathrm{M}(s) = \begin{bmatrix} -\frac{a_1(s)}{a_0(s)} & \dots & -\frac{a_{q-1}(s)}{a_0(s)} & -\frac{a_q(s)}{a_0(s)} \\ 1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & 0 \end{bmatrix} \tag{2.46}$$

Stability conditions of this type are routinely found in robust control theory, and are generally held as efficient measures in stability analysis. Unlike the classical results, it rids of any variable elimination procedure and lends a readily implementable criterion. The test can be easily checked by computing essentially the frequency-dependent measure $\rho(\mathrm{M}(j\omega)) < 1$, which is rather amenable to computation due to the ease in computing the spectral radius.

2.4.2 Stability Charts

Stability intervals can be extended to a two-dimensional (*2D*) map, known as a stability chart, in which the intervals are displayed with respect to a controller gain.

A stability chart can also be obtained in the plane of two delays, where each delay arises from a different input-output system in the closed-loop control. Compared to the one-dimensional (1D) stability analysis along a single delay axis, the stability information in a 2D delay plane is richer since it represents whether a system is stable or not with respect to all combinations of delays. A stability chart can reveal whether increasing a delay value favors stability or instability.

2.5 CHAPTER SUMMARY

In this chapter, the classical methods for TDS stability analysis were presented.

Firstly, it was shown that the stability tests are rooted in the same fundamental idea, which is to find the roots of characteristic equation in the complex plane. Then, the classical methods lied in a variable elimination procedure which required a complex symbolic computation.

Finally, the frequency sweeping method has been presented to provide sufficient conditions to have a DISS.

CHAPTER 3 NUMERICAL AND GRAPHICAL METHOD FOR THE STABILITY ANALYSIS OF COMMENSURATE MULTIPLE TIME-DELAY IMPERFECT SYSTEMS

3.1 CHAPTER OVERVIEW

In this chapter, by using the concepts covered in chapter 1 and 2, a numerical methodology which finds the coefficients of the characteristic quasi-polynomial is presented. Some case studied will be illustrated to show how the random number generation method affects the procedure convergence.

Then, the stability analysis to find the purely imaginary characteristic roots of LTI will be carried out through a graphical method. This method gives a quick conclusion on the stability analysis which means finding if the system is delay-dependent or independent.

Comparative cases study between this graphical method and the numerical one (*Schur-Cohn*) will be shown in order to demonstrate the strengths and weaknesses of the presented method. A part of this chapter has been published in [43, 44].

3.2 NUMERICAL PROCEDURE TO OBTAIN THE COEFFICIENTS OF THE PSEUDO-POLYNOMIAL EQUATION

In this section, a numerical procedure to obtain the coefficients of the pseudo-polynomial equation of a commensurate time-delay system is described.

3.2.1 Proposed Procedure

The main objective of this section is to describe how the coefficients of the pseudo-polynomial characteristic equation can be obtained by using a numerical interpolation procedure. For the sake of simplicity, let us consider the case where $k=2$ in the equation 1.20. The described method can be easily generalized to any value of k. In the considered case, the equation (1.20) can be written as:

$$\begin{aligned} a(s,z) = (a_{00}s^n + a_{01}s^{n-1} + \cdots + a_{0n})z^0 \\ +(a_{11}s^{n-1} + (a_{12}s^{n-2} + \cdots + (a_{1n})z^1 \\ +(a_{21}s^{n-1} + (a_{22}s^{n-2} + \cdots + (a_{2n})z^2 \\ +(a_{31}s^{n-2} + (a_{32}s^{n-3} + \cdots + (a_{3n})z^3 \\ + \cdots + (a_{kn})z^q \end{aligned} \tag{3.47}$$

The above equation has r unknown coefficients, a_{00}, a_{kn}, with r given by the following formula:

$$r = \sum_{i=1}^{n} r_0 + ki, \quad r_0 = n+1 \tag{3.48}$$

In order to find the unknown coefficients, the following system of r linear equations should be solved:

$$C = H.M \tag{3.49}$$

with C, $M \in \mathbb{R}^r$, $H \in \mathbb{R}^{rxr}$ defined as follow:

$$M = [a_{0n}, \ldots, a_{00}, a_{1n}, \ldots, a_{11}, \ldots, a_{kn}] \tag{3.50}$$

$$C = [c_1, \ldots, c_r] \tag{3.51}$$

where $H(s,\tau)$ is:

$$H = (H_0 | H_1 | \ldots | H_q) \tag{3.52}$$

$$H_i = \begin{pmatrix} s_1^{n-i} z_1^i & s_1^{n-i-1} z_1^i & \cdots & z_1^i \\ s_2^{n-i} z_2^i & s_2^{n-i-1} z_2^i & \cdots & z_2^i \\ \vdots & \vdots & \vdots & \vdots \\ s_r^{n-i} z_r^i & s_r^{n-i-1} z_r^i & \cdots & z_r^i \end{pmatrix}, i = 0, \cdots, q \tag{3.53}$$

The c_i, $i = 1, \cdots, r$ coefficients can be computed by using 1.18:

$$c_i = det(s_i I - A_0 - \sum_{k=1}^{m} A_k z_i^k) \tag{3.54}$$

According to the system 3.49, the vector M containing the constant coefficients of the pseudo-polynomial can be obtained as:

$$M = H^{-1} C \tag{3.55}$$

where H^{-1} is the inverse of H. The system 3.54 is solved by using a numerical procedure which is described below.

Let us choose r random values to compute s_i and to construct a vector $S \in \mathbb{R}^{1xr}$ with $i = 1, \cdots, r$. The corresponding z_i values are calculated for each s_i according to the formula $z = e^{-\tau s}$. The coefficients c_i in 3.54 can therefore be calculated. The matrix H in 3.53 can also be determined using s_i. This is very critical task because ill-conditioning of the matrix H should be avoided. That is not obvious, giving that the matrix H has been obtained using random vector S.

3.2.2 Procedure Summary

The proposed algorithm is summarized in the following points:

Step 1. Compute the index $q = nxk$.

Step 2. Find the index r in 3.48.

Step 3. Generate a random vector S of size r.

Step 4. Calculate the vector C using 3.54

Step 5. Find the matrix H using 3.53.

Step 6. Solve the system 3.55 to obtain the vector M.

3.2.3 Case Studies

In the next section, two examples will be presented to show how the choice of the random values affects the matrix H.

Ex 3.2.3.1 Let us consider the following LTI-TDS with $n = 2$ and $k = 1$.

$$\dot{x}(t) = A_0 x(t) + A_1 x(t - \tau) \tag{3.56}$$

where

$$A_0 = \begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix} \tag{3.57}$$

$$A_1 = \begin{pmatrix} 0 & 0 \\ -9 & -1.5 \end{pmatrix} \tag{3.58}$$

It can be easily verified to be stable for $\tau = 0$. The characteristic quasi-polynomial is obtained as

$$a(s,z) = (s^2 - s + 1) + (\frac{3}{2}s + 9)z \tag{3.59}$$

The coefficients of the characteristic equation (3.59), obtained by using symbolic calculus, are:

$$coef = [1 \quad -1 \quad 1 \quad \frac{3}{2} \quad 9] \tag{3.60}$$

The results obtained with the proposed procedure, with different random number generators, are shown below, outlining how this choice affects the matrix inevitability. The effect of the choice of the random vector S in step 3 is evaluated using different distribution ranges. The procedure is repeated over 100 times to access the stability of the solution with respect to different choices of the random number generator. The performance is evaluated by computing the real coefficients using the symbolic calculus and comparing the mean and variance values of the two solutions.

Exponential Distribution

In this section, the random numbers are generated from the exponential distribution in order to obtain the quasi-polynomial coefficients .

In *MATLAB* to generate random numbers from this distribution the function *exprnd(mu)* with mean *mu* can be used.

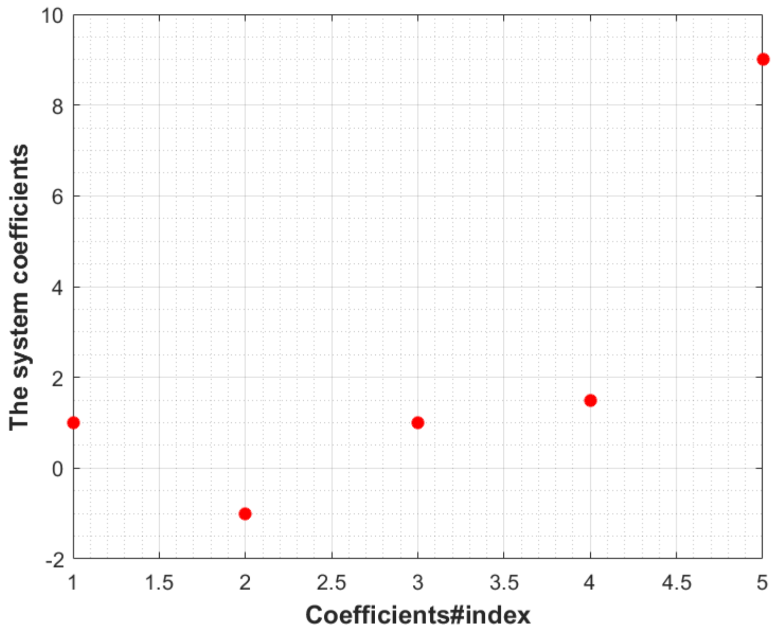


Figure 3.1: Comparison between the real coefficients (red) and the coefficients calculated from the exponential distribution (blue).

Figure. 3.1 shows that the coefficients generated using the numerical procedure and the symbolic one are equal for the 100 simulations run.

The mean value of the coefficients for the 100 simulations is:

$$coef_{mean} = [1 \quad -1 \quad 1 \quad \frac{3}{2} \quad 9] \tag{3.61}$$

with variance:

$$coef_{variance} = [1.75e-26 \quad 4.68e-24 \quad 5.12e-23 \quad 2.80e-21 \quad 5.53e-20] \tag{3.62}$$

The results shows that the exponential distribution is suitable to generate the random numbers to construct the quasi-polynomial coefficients.

The same results in term of mean values were obtained using the following distributions:

- Uniform Distribution
- Beta Distribution
- weibull distribution

It should be mention that the best results were obtained with exponential distribution for a very small variance. In the next example, the matrix system number and the time delays are increasing, then reporting how the coefficients are being affected using the above mentioned distributions.

Ex 3.2.3.2 Let us consider the following LTI-TDS with $n = 3$ and $k = 2$.

$$\dot{x}(t) = A_0 x(t) + A_1 x(t-\tau) + A_2 x(t-2\tau) \tag{3.63}$$

where

$$A_0 = \begin{pmatrix} 4 & -3 & 6 \\ 5 & 2 & 9 \\ 1 & 3 & 6 \end{pmatrix} \tag{3.64}$$

$$A_1 = \begin{pmatrix} 2 & 0 & 1 \\ -1 & 1 & 7 \\ 9 & 3 & 4 \end{pmatrix} \tag{3.65}$$

$$A_2 = \begin{pmatrix} 7 & -2 & 9 \\ -1 & 6 & 0 \\ 2 & -3 & 1 \end{pmatrix} \tag{3.66}$$

It can be easily verified to be stable at $\tau = 0$. The respective characteristic equation is

$$
\begin{aligned}
a(s,z) = (s^3 - 12s^2 + 26s - 81)\\
+(-7s^2 - 56s + 417)z\\
+(-14s^2 + 119s + 327)z^2\\
+(10s + 602)z^3\\
+(35s + 193)z^4\\
+217z^5\\
+41z^6
\end{aligned}
\tag{3.67}
$$

The coefficients of the equation (3.67), obtained by using symbolic calculus, are:

$$
\begin{aligned}
coef = [1 \quad -12 \quad 26 \quad -81 \quad -7 \quad -56 \quad 417 \quad -14\\
119 \quad 327 \quad 10 \quad 602 \quad 35 \quad 193 \quad 217 \quad 41]
\end{aligned}
\tag{3.68}
$$

Uniform Distribution

In this section, the random command in *MATLAB* is used for generating the values s_i in the interval *[0,1]*. Figure. 3.2 shows the solutions for the 100 simulations.

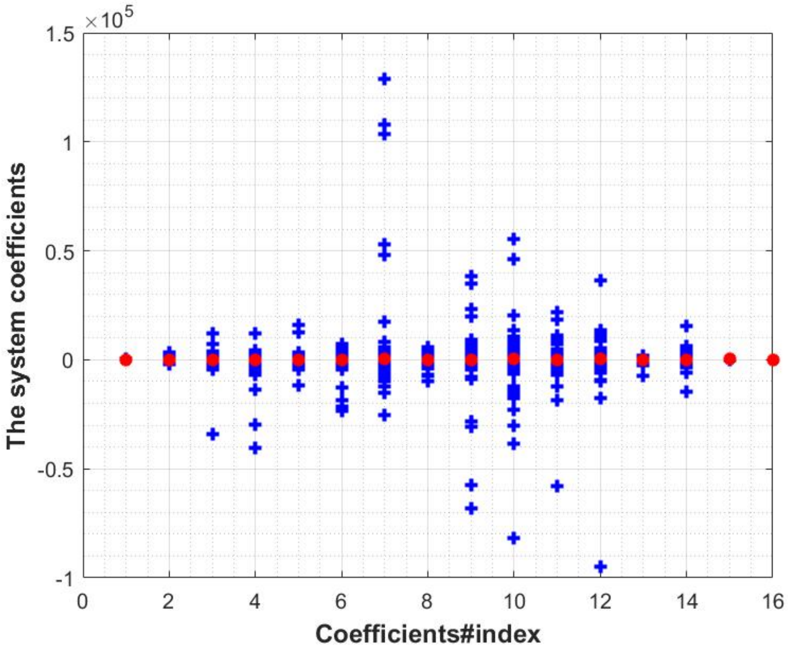


Figure 3.2: Comparison between the real coefficients (red) and the coefficients calculated by using the rand function (blue).

The mean value of the coefficients over the 100 simulations is:

$$coef_{mean} = [-1.6 \quad -55.71 \quad 732.03 \quad 781.43 \quad -913.23 \quad 2.104e3 \quad 2.7e3 \quad -8.5e3 \quad 1.25e4 \quad -7.15e3 \quad -9.27e3 \quad -914 \quad -2.41e3 \quad 177 \quad 42.62] \tag{3.69}$$

with variance:

$$coef_{varience} = [2.76e3 \quad 7.74e5 \quad 5.85e7 \quad 2.07e8 \quad 2.35e7 \quad 2.6e7 \quad 2.89e8 \quad 2.4e81.93e9 \quad 6.39e9 \quad 1.41e9 \quad 2.79e9 \quad 2.37e7 \quad 1.98e8 \quad 7.32e4 \quad 99.16] \tag{3.70}$$

It can be observed from Figure. 3.2 that the results of the rand function in terms of mean and variance for the case of two delays provided a large error between the values computed with respect to the symbolic one. Therefore the use of the rand function to generate the random numbers is not a suitable solution.

Exponential Distribution

In this section, the random numbers to generate the quasi-polynomial coefficients are from the exponential distribution.

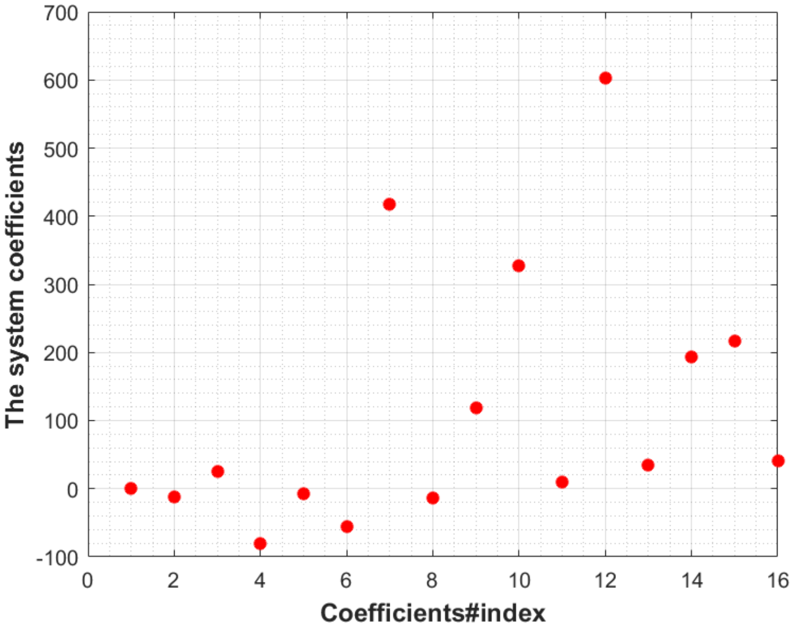


Figure 3.3: Comparison between the real coefficients (red) and the coefficients calculated by using the exponential distribution (blue).

The mean value over the 100 simulations is:

$$\begin{aligned} coef_{mean} = [1 \quad -12 \quad 26 \quad -81 \quad -7 \quad -56 \quad 417 \\ -14 \quad 119 \quad 327 \quad 10 \quad 602 \quad 35 \quad 193 \quad 217 \quad 41] \end{aligned} \tag{3.71}$$

$$\begin{aligned} coeff_{varience} = [8.44e-23 \quad 3.05e-19 \quad 1.25e-16 \quad 5.23e-16 \\ 3.11e-12 \quad 1.934e-9 \quad 7.277e-9 \quad 1.01e-6 \quad 2.1e-4 \quad 0.0017 \\ 0.0038 \quad 0.0046 \quad 0.2534 \quad 0.0076 \quad 0.0035 \quad 0.00152] \end{aligned} \tag{3.72}$$

As it can be observed from Figure. 3.2 that the exponential distribution provided a suitable results. The coefficients computed with the symbolic method and those obtained with this method are equal.

Weibull distribution

In this section, the wblrnd command in *MATLAB* will be used to generate the random numbers from the weibull distribution. Figure 3.4 shows the solutions over 100 simulations.

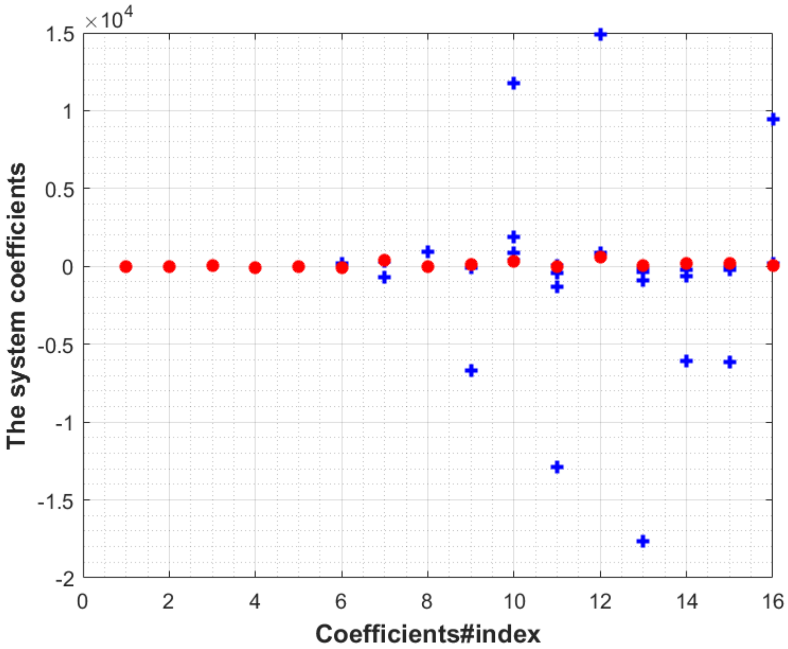


Figure 3.4: Comparison between the real coefficients (red) and the coefficients calculated by using the wblrnd function (blue).

The mean solution over *100* trials is:

$$coef_{mean} = [0.9996 \quad -11.98 \quad 25.82 \quad -80.56 \quad -10.16 \quad -26.45 \quad 297.76 \quad 82.32 \quad -593.49 \quad 1.68e3 \quad -1.45e3 \quad 2.06e3 \quad -1.87e3 \quad -562.45 \quad -492.37 \quad 1e3] \tag{3.73}$$

with variance of:

$$coef_{varience} = [6.027e-7 \quad 0.0015 \quad 0.2225 \quad 1.356 \quad 58.01 \quad 7.55e3 \quad 1.59e5 \quad 8.39e4 \quad 4.57e6 \quad 1.27e7 \quad 1.63e7 \quad 2.02e7 \quad 3.08e7 \quad 3.81e6 \quad 3.93e6 \quad 8.75e6] \tag{3.74}$$

Figure. 3.4 illustrates that the Weibull distribution is not suitable to generate the random numbers to obtain the coefficients of the quasi-polynomial.

The results described above show that the choice of the random number distribution is relevant to the robustness of the solution. Satisfactory results are obtained with a exponential distribution, which is able to guarantee the reliability and robustness of the method.

3.3 GRAPHICAL METHOD FOR DELAY DEPENDENT STABILITY ANALYSIS

Stability criteria based on frequency domain representations are time-honored tools in the study of dynamical systems. Classical examples of frequency domain stability criteria include different results such as the Nyquist test and root-locus method. With the aid of the small gain theorem, frequency domain tests have become increasingly more prevalent in stability analysis, and have played especially a central role in the theory of robust control. More generally, while frequency-domain methods are used predominantly in the analysis of linear systems, they have also found utilities in the studies of nonlinear systems. Various frequency-sweeping tests are now commonplace. This section develops a frequency domain stability method for LTI systems with commensurate delays. From Chapter 2, it is well known that the stability of an LTI delay system can be completely characterized by its characteristic roots. In this section, a graphical method to obtain the imaginary roots of the characteristic equation is described. Then, by using equation (1.22), the delay margin for which the system becomes unstable can be computed.

3.3.1 Graphical Method Procedure

As a first step, the characteristic equation introduced in (1.17) can be rewritten as:

$$det(j\omega - A_0 - \sum_{k=0}^{m} A_k z^k) = det(\lambda I - A) = 0 \tag{3.75}$$

where

$$A = A_0 + \sum_{k=0}^{m} A_k z^k \tag{3.76}$$

and $z = e^{-j\theta}$ and λ is the eigenvalues of the matrix A. It represents the imaginary roots of the characteristic equation. As described in Chapter 2, to obtain the imaginary roots, following variable definitions are taken into account $\lambda = j\omega$, $z = e^{-j\theta}$, where $\omega > 0$, and $\theta \in [0, 2\pi]$.

Therefore, the method is based on fixing a number r of θ_p values where $p = [1 \ldots r]$. It should be mentioned that the choice of r is very important because it impacts the precision of the eigenvalues and therefore their real parts; the higher is r value, the closer to zero is its real part.

The next steps are the computation of the matrix A for each value of θ_p, and then the computation and the plot of the eigenvalues λ_p by using a numerical algorithm, which has been implemented in *MATLAB*.

From the obtained plot, a conclusion can be done easily on the system stability analysis: if there are no crosses of the plot with the imaginary axes, it can be concluded that the system is delay independent stable, i.e. it is stable for any $\tau \in [0, \inf)$.

Otherwise, if the obtained plot presents crossing of the imaginary axes, the system stability is time-delay dependent and it exists a delay margin for which the system become unstable. The imaginary roots $\lambda_i = j\omega_i$ can be easily extracted along with the crossing frequencies ω_i to be used in the next step.

The last step is the computation of the delay margin, that can be done by computing the corresponding z_i over the crossing frequencies ω_i found in the previous step. It should be mentioned that $z_i = e^{-j\omega_i\tau}$ is a vector of solutions. The delay margin is calculated according to (1.22) and it presents the minimum value of the solution vector.

3.3.2 Algorithm to Find the Characteristic Equation Imaginary Roots

The procedure is summarized below. Given a system matrices with size n and time-delay k:

Step 1. Choose a value for r , and determine the r values of θ_p.

Step 2. Compute all the matrices A_p for each θ_p by using (3.75).

Step 3. For each matrix A_p, compute the corresponding eigenvalues vector λ_p.

Step 4. Plot all the eigenvalues λ_p on the complex plane $s = \sigma + j\omega$.

Step 5. From the plot, get its N crossing points on the imaginary axe ω_i.

Step 6. If $N \geq 1$, construct a vector $\omega_i \in [1, \ldots N]$ containing these values.

Step 7. Compute the vector z_i (size q) corresponding to the imaginary roots ω_i obtained in step 6 by using (3.75).

Step 8. Compute the delay margin by using (1.22).

The graphical method proposed in this section for the stability analysis of TDSs is a very feasible method, due to the directness of getting the imaginary roots. The method does not require the solution of high degree 2 variables polynomials, as described above for some of the analytical methods. Also, using the proposed graphical method enables us to visualize in short time a conclusion on the system stability analysis. A drawback of this graphical method is that it is not possible to have computed roots with the real parts perfectly equal to zero, although they are very close to the imaginary axe.

In this case, the criteria followed to choose the crossing ω_i frequencies is to search for their real part zero crossing. For an acceptable accuracy, a good practice is to set a value of r sufficiently high ($>$1000). This, in turn, is more demanding in terms of execution speed and RAM memory size. In the next section, numerical examples are reported to show the behavior of the proposed method.

3.3.3 Case Studies

In this part, three examples are reported to show a comparison among the proposed method and the analytical method (Schur-Cohn).

Ex 3.3.3.1 Let us consider the following LTI-TDS with $n = 2$ and $k = 1$.

$$\dot{x}(t) = A_0 x(t) + A_1 x(t - \tau) \tag{3.77}$$

with the system matrices defined as:

$$A_0 = \begin{pmatrix} -4 & -1 \\ 3.57 & 0 \end{pmatrix} \tag{3.78}$$

$$A_1 = \begin{pmatrix} -6 & -1 \\ 3 & -2 \end{pmatrix} \tag{3.79}$$

The eigenvalues of the system matrices are:

- $\lambda_{A0} = -6.4142, -3.5858,$
- $\lambda_{A1} = -5, -3$
- $\lambda_{A_0+A_1} = -9.0000 + 1.7321i, -9.0000 - 1.7321i.$

Therefore, the system (3.77) is stable for $\tau = 0$. The corresponding characteristic equation for this system is:

$$\begin{aligned} a(s,z) = s^2 + 10s + 23 \\ +(8s + 46)z \\ +15z^2 \end{aligned} \tag{3.80}$$

Let us use the Schur-Cohn method and make the computation of the imaginary roots of (3.77), it they exist.

To this aim, the method described in the Section above was adopted and the computation of the matrix $\sum(s)$ and the polynomial $det(\sum(s))$ was done.

The results of the computation show that:

- the system stability is time-delay dependent;
- the matrix $\sum(s)$ size is 4×4

$$S = \begin{pmatrix} s^2+10s+23 & 0 & 15 & 8s+46 \\ 8s+46 & s^2+10s+23 & 0 & 15 \\ 15 & 0 & s^2-10s+23 & 46-8s \\ 46-8s & 15 & 0 & s^2-10s+23 \end{pmatrix} \tag{3.81}$$

- The polynomial $det(\sum(s))$ degree is $2n^2 = 50$ and the imaginary root is:

$$\sigma \pm j\omega = \pm j4.2937 \tag{3.82}$$

The Table (3.1) illustrates the delay corresponding to the crossing frequency of this system.

Now the crossing frequencies in the case of the graphical method are evaluated.

Table 3.1: Delays corresponding to the crossing frequencies of the system (3.77) using Schur-Cohn method

Crossing frequency ω_k	$z = e^{(-j\omega_k \tau)}$	Time delay τ_k
4.2937	-0.606-j0.795	0.5173

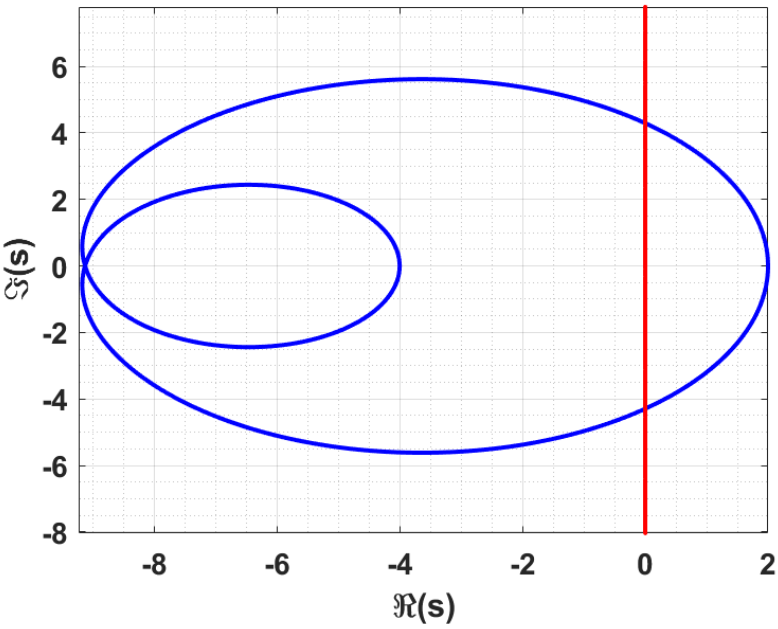


Figure 3.5: Plot of the eigenvalues of the characteristic equation (3.80) in the complex plane of the system (blue) and the imaginary axis (red) for the system 3.77.

From Figure. 3.5, only one imaginary root that cross the imaginary axis is observed, where:

$$\sigma \pm j\omega = \pm j4.3 \tag{3.83}$$

The correspondent delay margin is $\tau = 0.518$.

Ex 3.3.3.2 This part presents the second example in which the system has a single time delay. Let us consider the following LTI-TDS with $n = 5$

and $k = 1$.

$$\dot{x}(t) = A_0 x(t) + A_1 x(t - \tau) \tag{3.84}$$

with the system matrices defined as:

$$A_0 = \begin{pmatrix} -2 & -0.9 & 1.1 & -0.075 & 0 \\ 3.57 & 0 & -1.25 & 0 & -2 \\ -0.23 & 0.5 & 0 & -0.16 & 2 \\ 0 & 0 & 0.16 & 0 & 0 \\ -3 & 0 & -1 & 0.265 & -1.86 \end{pmatrix} \tag{3.85}$$

$$A_1 = \begin{pmatrix} -3 & -2.67 & 0 & 0 & 0 \\ 0.89 & 0 & -1.2 & -3 & -2 \\ -0.7 & 0.2 & 0 & -0.16 & 2 \\ 0 & 0 & 0.9 & 0 & 0 \\ -5 & 3.8 & -1 & 2.9 & -3 \end{pmatrix} \tag{3.86}$$

According to (1.17) it can be easily verified that the system matrices are stable and, therefore, the system is stable for $\tau = 0$.

The corresponding characteristic equation for the system is:

$$
\begin{aligned}
a(s,z) = {} & s^5 + 3.86s^4 + 9.8366s^3 + 22.1658s^2 + 6.6107s + 0.197 \\
& +(5.982s^4 + 32.6665s^3 + 114.1315s^2 + 57.3834s + 4.9852)z \\
& +(18.7063s^3 + 158.9594s^2 + 176.7873s + 43.3706)z^2 \\
& +(61.3260s^2 + 195.8586s + 163.7519)z^3 \\
& +(71.2247s + 263.7726)z^4 + (139.8779)z^5
\end{aligned} \tag{3.87}
$$

Let us use the Schur-Cohn method and make the computation of the imaginary roots of (3.84), it they exist. To this aim, the method described in the Section above was followed and the computation of the matrix $\sum(s)$ and the polynomial $det(\sum(s))$ was done.

The results of the computation show that:

- the system stability is time delay dependent;
- the matrix $\sum(s)$ size is 10×10;
- the polynomial $det(\sum(s))$ degree is $2n^2 = 50$ and the imaginary roots are:

$$\sigma \pm j\omega = \pm j6.1133, \pm j3.0060, \pm j1.3618, \pm j0.8398 \tag{3.88}$$

The table below presents the delays corresponding to the crossing frequencies of this system:

from Table 3.2 $\tau_{min} = 0.0315$ is the delay margin. This next part focuses on the computing of the imaginary roots using the graphical

Table 3.2: Delays corresponding to the crossing frequencies of the system (3.84) using Schur-Cohn method

Crossing frequency ω_k	$z = e^{(-j\omega_k\tau)}$	Time delay τ_k
6.1133	0.9816 -j0.192	0.03153
3.0060	0.0965 -j0.088	0.4904
1.3618	-0.23 - j0.040	1.3239
1.3221	-0.235- j0.0389	1.3676
0.8398	-0.282 -j0.0153	2.2119

method. As described in Section above the graphical method is based on finding the eigenvalues:

$$det(\lambda_p I - A_p) = 0 \text{where} \quad A_p = A_0 + A_1 z_p, p \in [0 \ldots 6284], \qquad (3.89)$$

and $\omega_i = 0 : 0.001 : 2\pi$..

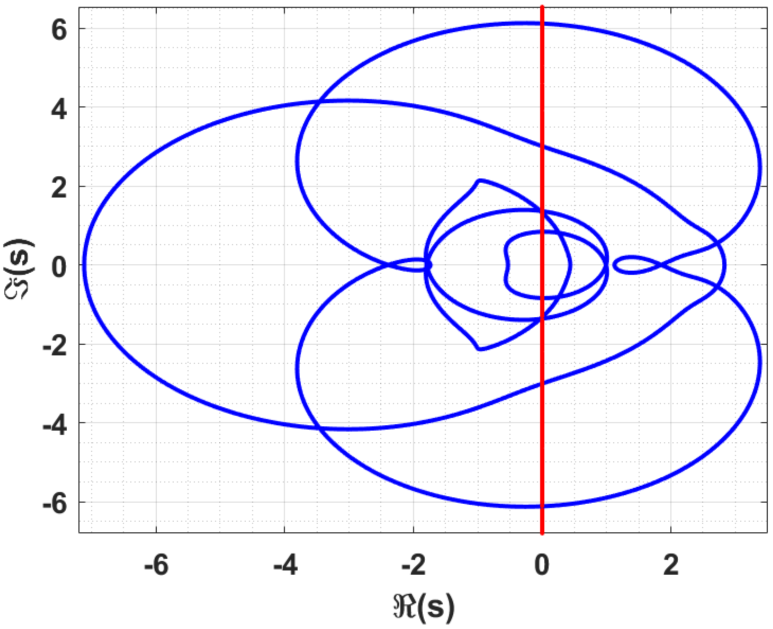


Figure 3.6: Plot of the eigenvalues of the characteristic equation in the complex plane of the system (blue) and the imaginary axis (red) for the system 3.84.

Based on the Figure. 3.6, the curve crosses the imaginary axes. It can be concluded that the system (3.84) stability is delay dependent. The plot shows that exist five crossing points and they are the imaginary roots of (3.84):

$$\sigma \pm j\omega = \pm j6.1133, \pm j3.0070, \pm j1.362, \pm j0.8398 \tag{3.90}$$

According to the steps described in the Section above, it is straightforward to obtain the corresponding z for each frequency crossing (3.90). The following Table **??**, presents the z and the time delay τ_k corresponding to the crossing frequencies.

According to Table. **??** $\tau_{min} = 0.0314$ is the delay margin. The res-

Table 3.3: Delays corresponding to the crossing frequencies of the system (3.84) using graphical method

Crossing frequency ω_k	$z = e^{(-j\omega_k\tau)}$	Time delay τ_k
6.1133	0.9815 -j0.191	0.0314
3.0070	0.096 -j0.088	0.4902
1.362	-0.23 - j0.040	1.3237
1.3221	-0.235- j0.0389	1.3676
0.8398	-0.282 -j0.0153	2.2119

ults of the two methods are almost equal in term of the imaginary roots and the delay margin. As expected, the example demonstrates that the graphical method gives an immediate result in term of stability analysis and imaginary roots.

Ex 3.3.3.3 This part presents a third example in which the system has four time delays. Let us consider the following LTI-TDS with $n = 3$ and $k = 4$.

$$\dot{x}(t) = A_0 x(t) + A_1 x(t-\tau) + A_2 x(t-2\tau) + A_3 x(t-3\tau) + A_4 x(t-4\tau) \tag{3.91}$$

with the system matrices defined as:

$$A_0 = \begin{pmatrix} -1 & 4.5 & -7 \\ -2.5 & 0 & -5.08 \\ 0 & 1 & -4 \end{pmatrix} \tag{3.92}$$

$$A_1 = \begin{pmatrix} -4.8 & 5 & -2.4 \\ -1.3 & -3.2 & 0 \\ -2.7 & 1 & -6 \end{pmatrix} \tag{3.93}$$

$$A_2 = \begin{pmatrix} -6 & -33 & -9 \\ 6.7 & -1 & 5 \\ 1 & 3 & -2 \end{pmatrix} \tag{3.94}$$

$$A_3 = \begin{pmatrix} -8.9 & -27 & -18.9 \\ 2 & 1 & 7 \\ 1 & -2 & 4 \end{pmatrix} \tag{3.95}$$

$$A_4 = \begin{pmatrix} -5 & 2 & -3.4 \\ 1 & -2 & 0 \\ -5.9 & 0 & 2 \end{pmatrix} \tag{3.96}$$

It can simply verified that the system matrices are stable, and therefore the system is stable for $\tau = 0$.

The corresponding characteristic equation for this system is:

$$\begin{aligned} a(s,z) = {} & s^3 + 5s^2 + 20.33s + 32.58 + (14s^2 + 45.73s + 88.842)z \\ & + (9s^2 - 1.03s - 294.776)z^2 + (3.9s^2 - 81.36s - 431.044)z^3 \\ & + (5s^2 + 243.25s + 337.68)z^4 + (308.68s + 165.738)z^5 \\ & + (139.5s + 1109.836)z^6 + (106.91s + 130.914)z^7 \\ & + (14.06s + 553.876)z^8 + (1751.602)z^9 + (836.25)z^{10} \\ & + (16.76)z^{11} + (24.12)z^{12} \end{aligned}$$

A computation of the polynomial $det(\sum(s))$ was done with the Schur-Cohn method. The results show that:

- The system stability is time delay dependent;
- The size of the matrix $det(\sum(s))$ is $24x24$;
- The polynomial $det(\sum(s))$ degree is $2n^2 = 72$ and the imaginary roots are:

$$\sigma \pm j\omega = \pm j30.029, \pm j19.446, \pm \\ j16.956, \pm j11.905, \pm j8.4169, \pm j1.8344, \pm j1.2601$$

The Table. 3.4 presents the delays corresponding to the crossing frequencies of this system: According to Table. 3.4 $\tau_{min} = 0.0555$ is the delay margin. In the next part the graphical method is used for computing the imaginary roots. The *Figure 3.7* below represents the plot eigenvalues of the matrix A.

$$det(\lambda_p I - A_p) = 0; \quad p \in [0\ldots 6284] \quad \text{where} A_p = A_0 + A_1 z_p + A_2 z_p^2 + A_3 z_p^3 + A_4 z_p^4 \tag{3.97}$$

Table 3.4: Delays corresponding to the crossing frequencies of the system (3.91) using Schur-Cohn method

Crossing frequency ω_k	$z = e^{(-j\omega_k\tau)}$	Time delay τ_k
30.029	0.095-j1.014	0.0555
19.446	-0.052-j0.796	0.0835
16.956	-0.0413-j0.729	0.0951
11.905	-0.0239-j0.5606	0.134
8.4169	-0.029-j0.405	0.1901
1.8344	-0.205-j0.074	0.9693
1.2601	-0.210-j0.0565	1.4148

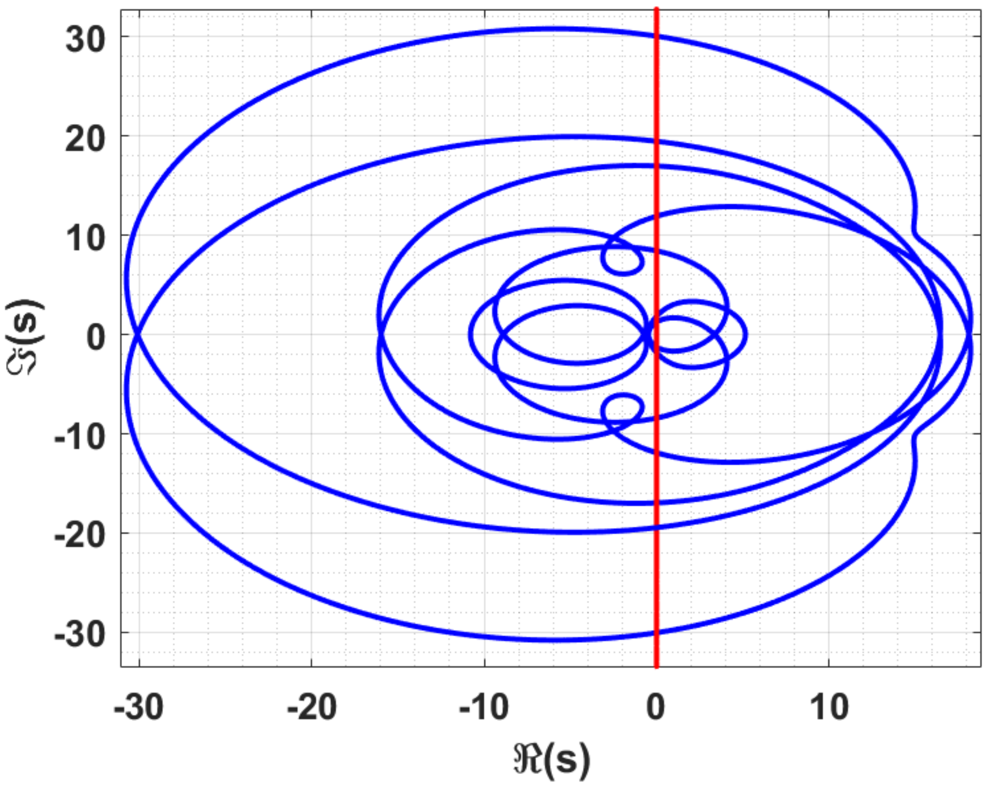


Figure 3.7: Plot of the eigenvalues of the characteristic equation in the complex plane of the system (blue) and the imaginary axis (red) for the system 3.91.

From the Figure. 3.7 the imaginary roots can be get easily:

$$\sigma \pm j\omega = \pm j30.029, \pm j19.446, \pm$$
$$j16.956, \pm j11.905, \pm j8.4169, \pm j1.8344, \pm j1.2601$$

The delays corresponding to the crossing frequencies of this system are:

Table 3.5: Delays corresponding to the crossing frequencies of the system (3.91) using graphical method

Crossing frequency ω_k	$z = e^{(-j\omega_k\tau)}$	Time delay τ_k
30.03	0.095-j1.014	0.0555
19.58	-0.053-j0.799	0.0829
16.96	-0.0414-j0.729	0.0951
11.91	-0.0239-j0.5608	0.1339
8.4169	-0.029-j0.405	0.1901
1.8344	-0.205-j0.074	0.9695
1.26	-0.210-j0.0566	1.4150

According to Table 3.5, τ_{min} is the delay margin.

In addition to the comments written above for the comparison in first example, this one shows that the graphical method does not limit to a single delay, but it can be extended to an infinite commensurate time delays, thus demonstrating the robustness of the method proposed.

3.4 CHAPTER SUMMARY

A method to obtain the coefficients of the quasi-polynomial has been developed. The method can be used to substitute the symbolic method that is time-consuming when we have multiple commensurate time-delay systems. The method is based on constructing a new matrices system to find the coefficients of pseudo-polynomial and it is based on generating a suitable random numbers to find the system solution. Then, a graphical method for the stability analysis of TDSs with commensurate multiple time delays has been presented. The results of this study indicate that the graphical method is an efficient alternative approach for the stability time delay systems analysis. Satisfactory results have been obtained by showing all the imaginary roots in a graphical plot.

CHAPTER 4 DELAY INDEPENDENT STABILITY CONTROL FOR COMMENSURATE MULTIPLE TIME-DELAY SYSTEMS

4.1 CHAPTER OVERVIEW

The major difficulty in the field of TDSs is the study of the asymptotic stability with respect to the delay. The stability analysis of TDSs is not a trivial task, due to the infinite number of roots in the complex plane of the characteristic equation associated to the TDSs. Moreover, the system could be stable independently of the delay value DISSs or it could be stable only for some values of the delay DDSSs. A DDE that is stable for only some values in the delay parameter space is called DDSS. Otherwise if the stability of a DDE is maintained independently of the delay, then DDE is called DISS.

Multiple disjoint delay regions may also exist where the system may be stable within each region, while becoming unstable outside. These regions, which are known as stability regions, become stability intervals.

Besides, there exist several methods for clustering the delay parameter space based on the τ-decomposition concept. It consists in dividing the delay region into intervals with the same number of unstable roots NU(τ). For example, in [50] the stability regions with respect to the delay are determined, and a method called cluster treatment of characteristic roots (CTCR) is proposed. It can detect all the stability regions of a DDSS in the space of the time delays. The study is focused on finding specific intervals for the time delays for which the system is stable and has been developed only for third-order systems with two time-delay parameters.

In this chapter, we introduce two approaches that allow to make the system DISS. The first method is based on the design of a controller which is based on a single gain. The control gain parameter that allows shifting the roots from the Right Half complex Plane (RHP) to the Left Half Plane (LHP) for any value of delay is therefore derived. The second method uses two parameters to make the system DISS. Both approaches can be applied to any commensurate multiple TDS (CMTDS), independently from the number of delays.

This chapter is organized as follows: Section 4.2 introduces the new design procedure to obtain a DISS and the proof of the related theorem; Section 4.3 presents two algorithms allowing to get the suitable controller gain parameter; Section 4.4 shows the efficiency of the proposed approach through the discussion of several case studies. Finally, in Section 4.5 the main results for DISS using two parameters will be presented.

4.2 A LINEAR STATE FEEDBACK REGULATOR TO MAKE SYSTEMS DELAY-INDEPENDENT STABLE

In this section, the main design procedure will be described. Let us consider a CMTDS as in Eq. (1.16) with A_0 and $\mathrm{A}_0 + \sum_{k=1}^{m} \mathrm{A}_k$ stable matrices. It exists a linear state-space regulator with gain matrix L_0 such that the controlled system is delay-independent stable, it is expressed as

$$\dot{\mathrm{x}}(t) = \bar{\mathrm{A}}_0 \mathrm{x}(t) + \sum_{k=1}^{m} \mathrm{A}_k \mathrm{x}(t - k\tau) \qquad (4.98)$$

where $\bar{\mathrm{A}}_0 = \mathrm{A}_0 - \mathrm{L}_0$ and $\mathrm{L}_0 = l\mathrm{I}$, with l being the scalar controller parameter gain and I is the identity matrix.

Proof. Let us consider the quasi-polynomial equation correspondent to the system (4.98), which can be written as:

$$a(s, e^{-\tau s}) = \det\left(s\mathrm{I} - \bar{\mathrm{A}}_0 - \sum_{k=1}^{m} \mathrm{A}_k e^{-k\tau s}\right) \tag{4.99}$$

that, in terms of the coefficients, becomes:

$$a(s,z) = \bar{a}_0(s) + \sum_{k=1}^{q} a_k(s) z^{-k} \qquad z = e^{\tau s} \tag{4.100}$$

with $\tau \geq 0$ and $\bar{a}_0(s), \ldots, \bar{a}_k(s)$ are the coefficients of the characteristic equation of the controlled system.

The frequency sweeping method, described in Chapter 2, is used to prove the existence of the parameter l. According to Lemma described in 4.2, the system (4.98) is stable if:

$$\rho(\bar{\mathrm{M}}(j\omega)) < 1 \tag{4.101}$$

where $\bar{\mathrm{M}}(s)$ is:

$$\bar{M}(s) = \begin{bmatrix} -\frac{a_1(s)}{\bar{a}_0(s)} & \ldots & -\frac{a_{m-1}(s)}{\bar{a}_0(s)} & -\frac{a_m(s)}{\bar{a}_0(s)} \\ 1 & \ldots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 1 & 0 \end{bmatrix} \tag{4.102}$$

for $s = j\omega$. The characteristic polynomial of matrix (4.102), whose structure follows that of a state matrix in a canonical form, can be written as:

$$\bar{a}_0(s)\lambda^m + a_1(s)\lambda^{m-1} + \ldots + a_m(s) = 0 \tag{4.103}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_m$ are the eigenvalues of $\bar{\mathrm{M}}$.

If the maximum modulus of $\lambda_1, \lambda_2, \ldots, \lambda_m$ is less than 1, $\bar{a}_0$ $(j\omega)$ is suitable, i.e. $\bar{a}_0(j\omega)$ makes the system DIS. By using the Rouché's theorem [55], the roots of the polynomial (4.103) are in modulus all less of the quantity

$$R = \max\left\{1, \frac{1}{|\bar{a}_0(j\omega)|}\left[|a_m(j\omega)| + |a_{m-1}(j\omega)| + \ldots + |a_1(j\omega)|\right]\right\} \tag{4.104}$$

Remind that, according to Eqs.(1.4), the degree of $\bar{a}_0(j\omega)$ in ω is always greater than $a_k(j\omega)$. Therefore, the $\bar{a}_0(j\omega)$ can be chosen in such a way that each term
$|\frac{a_m(j\omega)}{\bar{a}_0(j\omega)}|, |\frac{a_{m-1}(j\omega)}{\bar{a}_0(j\omega)}|, \ldots, |\frac{a_1(j\omega)}{\bar{a}_0(j\omega)}|$ is less than one. In order to make (4.101) true, i.e. $R < 1$, the $\bar{a}_0(j\omega)$ can be set in such a way that each term in (4.104) is less than $\frac{1}{m}$, which means the sum is less then 1 for all ω. This can be done acting just on $\bar{a}_0(j\omega)$, i.e. working on $\bar{\mathrm{A}}_0 = \mathrm{A}_0 - l\mathrm{I}$ the system becomes DIS. □

The value of l can be derived with different algorithms, as described below. The technique is a clear advance on current methods, since it allows us to stabilize a DDS with an infinite number of commensurate delays.

4.3 ALGORITHMS TO DETERMINE THE VALUE OF THE CONTROL GAIN

In this section, two algorithms will be presented to get a suitable control parameter $\bar{l}$.

4.3.1 Algorithm 1

The characteristic equation of the controlled system is $B(s,l)$ as in Eq. (2.38), which is a function of s and l. The condition in which the characteristic equation admits a pair of purely imaginary solutions should be found by varying l. Due to the continuity of the roots of $B(s,l)$ with respect to l, this is the limit condition for roots passing from the RHP to the LHP. The condition can be retrieved performing the polynomial division between $B(s,l)$ and a generic polynomial $B_1(s) = s^2 + \alpha$, with $\alpha > 0$, and searching the values of α and l for which the remainder is null, i.e. $R(\alpha, l) = 0$.

The algorithm is based on fixing different values of α and calculating the roots of $R(\alpha, l)$ with respect to l. The value $\bar{l}$ is chosen as $\max_l \{R(\alpha,l) = 0, \forall \alpha\}$.

An alternative way to find $\bar{l}$ is computing the roots of the polynomial $B(s,l)$ for a high value of l and decreasing the parameter l until all roots are in the LHP. A precise value of $\bar{l}$ can be obtained applying the bisection algorithm.

4.3.2 Algorithm 2

In the case of a low order polynomial $B(s,l)$, the Routh criterion can be adopted to find an analytical expression for the stabilizing controller parameter l.

The method relies on finding the values of l for which null entries on the first column of the Routh table appear, thus ensuring the existence of purely imaginary roots for $B(s,l)$. Considering that $B(s,l)$ is an even polynomial of degree n, a variable change $t = s^2$ leads to the polynomial $D(t,l)$ of degree $\frac{n}{2}$. The Routh table for $D(t,l)$ is constructed and the roots of $C_i(l)$ can be found, where $C_i(l), i = 1, \ldots, \frac{n}{2}$ are the terms of the first column of the Routh table. In this way, these values of l ensure the existence of two purely imaginary roots $D(t,l)$. Substituting the real positive roots of $C_i(l)$ in polynomial $D(t,l)$ leads to factorization of type $D(t) = F(t)(t-t_1)\cdot(t-t_{n/2-2})$, where $F(t)$ is the polynomial associated to the two imaginary roots and $t_1, \ldots, t_{n/2-2}$ are the real solutions of $D(t)$. When $t_i < 0$ for a given i, a real negative root for $D(t)$ is found and, hence, a pair of purely imaginary roots for $B(s,l)$. The controller gain parameter represents the largest value of l satisfying this condition.

4.4 ILLUSTRATIVE EXAMPLES

In this section, three illustrative examples are presented.

Ex 4.4.0.1 Let us consider the system presented in [28], where a single delay τ occurs:

$$\dot{\mathrm{x}}(t) = \mathrm{A}_0\mathrm{x}(t) + \mathrm{A}_1\mathrm{x}(t-\tau) \tag{4.105}$$

with the system matrices defined as:

$$\mathrm{A}_0 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2\pi-\pi^2 & 2 \end{pmatrix} \tag{4.106}$$

$$\mathrm{A}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \pi^3 & -2\pi & 2-\pi \end{pmatrix} \tag{4.107}$$

The roots of the characteristic function are plotted in blue in Figure. 4.1, retrieving that the system is DDS, since the curves cross the imaginary axis.

Method 1. By applying the algorithm presented in Subsect. 4.3.1, the matrix $\bar{\mathrm{A}}_0$ is defined as:

$$\bar{\mathrm{A}}_0 = \mathrm{A}_0 - \mathrm{L}_0 = \begin{bmatrix} 0-l & 1 & 0 \\ 0 & 0-l & 1 \\ 0 & -2\pi-\pi^2 & 2-l \end{bmatrix} \tag{4.108}$$

where l is the control parameter. The polynomial $B(s,l)$ is computed and divided by $B_1(s) = s^2+\alpha$, obtaining the remainder $R(\alpha,l)$. Then, different values of α are fixed, i.e. $\bar{\alpha} = 1,\ldots,100$, and the positive real roots of the polynomial $R(\bar{\alpha},l)$ are plotted, as reported in Figure. 4.2. The maximum value $\bar{l}$ in the graph represents the controller gain parameter

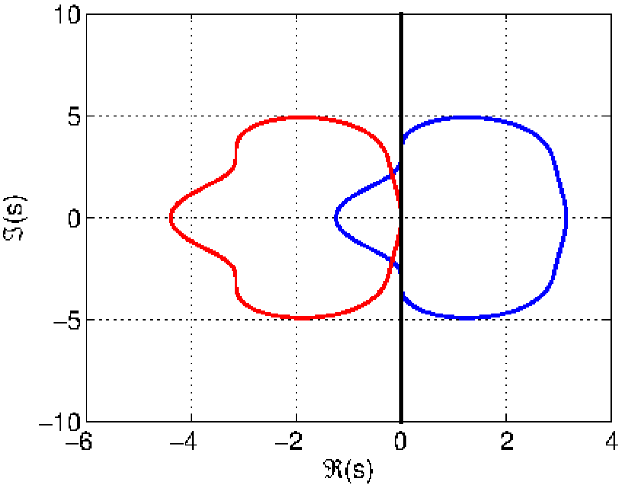


Figure 4.1: Solutions of Eq. (1.17) (in blue) and of Eq. (4.99) with $\bar{l} = \pi$ for Example .4.4.0.1

that makes the system DIS, i.e. $\bar{l} = \pi$, as shown in red in Figure. 4.1.

Alternatively, $\bar{l}$ can be found by fixing $l = 10$, verifying graphically that all roots of $B(s,l)$ are in the LHP and decreasing l until the roots cross the imaginary axis, or by using the bisection algorithm. Then, again $\bar{l} = \pi$.

Method 2. The value $\bar{l}$ is derived analytically by applying the method in Subsect. 4.3.2. The polynomial $B(s,l)$ is calculated as:

$$\begin{aligned} &B(s,l) = -s^6 + (3l^2 - 4l - 29.6088)s^4 + \\ &(-3l^4 + 8l^3 - 5.3935l^2 + 78.9568l - 292.2273)s^2 + \\ &+(L^6 - 4l^5 + 35.0023l^4 - 78.9568l^3 + 150.6411l^2 - \\ &-389.6364l - 961.3892) \end{aligned} \tag{4.109}$$

and, assuming $t = s^2$, the polynomial $D(t,l)$ is obtained as:

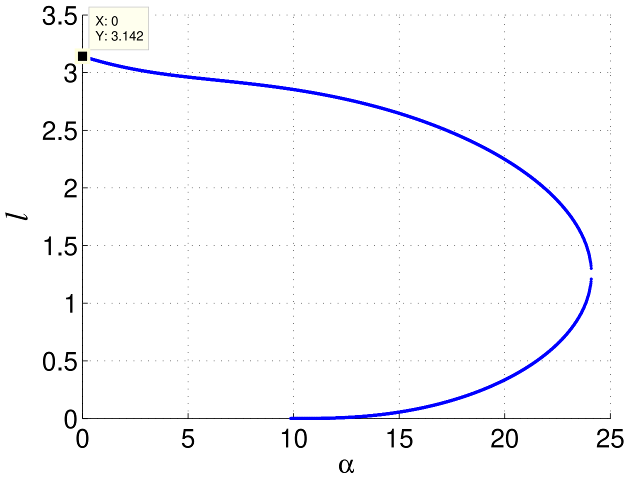


Figure 4.2: Positive real solutions of $R(\bar{\alpha}, l)$. The value of $\bar{l}$ is the maximum, i.e. $\bar{l} = \pi$.

t^3	-1	$-3l^4+8l^3-5.4l^2+79l-292.2$
t^2	$3l^2-4l-29.6$	$l^6-4l^5+35l^4-79l^3+150.6l^2-389.6l-961.4$
t^1	$(8l^6-32l^5-75.6l^4+57.4l^3+882.2l^2+1558.5l-7691.1)/(3l^2-4l-29.6088)$	0
t^0	$l^6-4l^5+35l^4-79l^3+150.6l^2-389.6l-961.4$	0

Table 4.1: Routh table for polynomial $D(t,l)$ as in Eq.(4.110).

$$\begin{aligned} D(t,l) &= -t^3+(3l^2-4l-29.6088)t^2+ \\ &(-3l^4+8l^3-5.3935l^2+78.9568l-292.2273)t+ \\ &+(l^6-4l^5+35.0023l^4-78.9568l^3+150.6411l^2 \\ &-389.6364l-961.3892) \end{aligned} \tag{4.110}$$

The Routh table for Eq. (4.110), with l parameter, is constructed as reported in Tab. 4.1.

The values of l for which the elements of the first column of the Routh table are null, imply the existence of purely imaginary solutions for $D(t,l)$. Therefore, for these values of l, the polynomial $D(t)$ can be factorized as $D(t) = F(t)(t+t_1)\cdot(t+t_{n/2-2})$. Thus, the controller gain parameter ensuring the DIS for system (4.105) can be chosen as the maximum value of l which makes null one (or more) elements in the first column of the Routh table. In this case, $\bar{l} = \pi$ is one of the solutions of the polynomial in l appearing as the last element of the first column of the Routh table in Tab. 4.1.

Ex 4.4.0.2 Let us consider the fifth-order TDS, with one delay parameter and system matrices:

$$\mathrm{A}_0 = \begin{pmatrix} -2 & -0.9 & 1.1 & -0.075 & 0 \\ 3.57 & 0 & -1.25 & 0 & -2 \\ -0.23 & 0.5 & 0 & -0.16 & 2 \\ 0 & 0 & 0.16 & 0 & 0 \\ -3 & 0 & -1 & 0.265 & -1.86 \end{pmatrix} \tag{4.111}$$

$$\mathrm{A}_1 = \begin{pmatrix} -3 & 2.67 & 0 & 0 & 0 \\ 0.89 & 0 & -1.2 & -3 & -2 \\ -0.7 & 0.2 & 0 & -0.16 & 2 \\ 0 & 0 & 0.9 & 0 & 0 \\ -5 & 3.8 & -1 & 2.9 & -3 \end{pmatrix} \tag{4.112}$$

The system is DDS as shown in the blue curves in Figure. 4.3 which represents the roots of the characteristic function.

Method 1. The controlled matrix $\bar{\mathrm{A}}_0$ is:

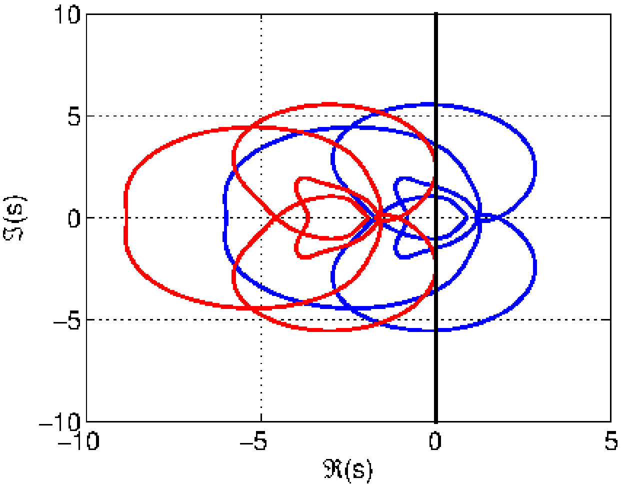


Figure 4.3: Solutions of Eq. (1.17) (in blue) and of Eq. (4.99) with $\bar{l} = 2.8384$ for Example 4.4.0.2

$$\bar{\mathrm{A}}_0 = \mathrm{A}_0 - \mathrm{L}_0 = \begin{pmatrix} -2-l & -0.9 & 1.1 & -0.075 & 0 \\ 3.57 & -l & -1.25 & 0 & -2 \\ -0.23 & 0.5 & 0-l & -0.16 & 2 \\ 0 & 0 & 0.16 & 0-l & 0 \\ -3 & 0 & -1 & 0.265 & -1.86-l \end{pmatrix} \tag{4.113}$$

The controller gain $\bar{l} = 2.8384$ is found by following the procedure described above as reported in the plot in Figure. 4.4. The control is effective as shown by the roots of the characteristic polynomial in Eq. (1.17) reported in red in Figure. 4.4. Similar results can be found using the alternative procedure of algorithm in Subsect. 4.3.1. In this case, the application of algorithm in Subsect. 4.3.2 would lead to an high order $B(s,l)$ polynomial and hence to a more complex Routh table.

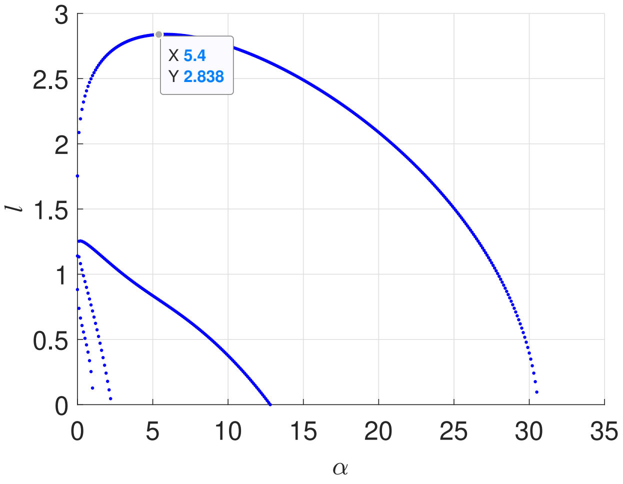


Figure 4.4: Positive real solutions of $R(\bar{\alpha}, l)$. The value of $\bar{l}$ is the maximum, i.e. $\bar{l} = 2.8384$.

Ex 4.4.0.3 Consider the following TDS, with third-order dynamics and three delay parameters as described in compact form as:

$$\dot{\mathrm{x}}(t) = \mathrm{A}_0\mathrm{x}(t) + \mathrm{A}_1\mathrm{x}(t-\tau) + \mathrm{A}_2\mathrm{x}(t-2\tau) + \mathrm{A}_3\mathrm{x}(t-3\tau) \tag{4.114}$$

with system matrices defined as:

$$\begin{array}{ll} \mathrm{A}_0 = \begin{pmatrix} -1 & 4.5 & -7 \\ -2.5 & 0 & -5.08 \\ 0 & 1 & -4 \end{pmatrix} & \mathrm{A}_1 = \begin{pmatrix} -4.8 & 5 & -2.4 \\ -1.3 & -3.2 & 0 \\ -2.7 & 1 & -6 \end{pmatrix} \\ \mathrm{A}_2 = \begin{pmatrix} -6 & -33 & -9 \\ 6.7 & -1 & 5 \\ 1 & 3 & -2 \end{pmatrix} & \mathrm{A}_3 = \begin{pmatrix} -8.9 & -27 & -18.9 \\ 2 & 1 & 7 \\ 1 & -2 & 4 \end{pmatrix} \end{array} \tag{4.115}$$

It can be verified that the system matrices satisfy hypotheses of Theorem 4.2, while Figure. 4.5 illustrates that the system is DDS.

Method 1. Similarly to what have been done in the previous examples, from Figure. 4.6, the controller gain $\bar{l} = 17.86$ can be retrieved

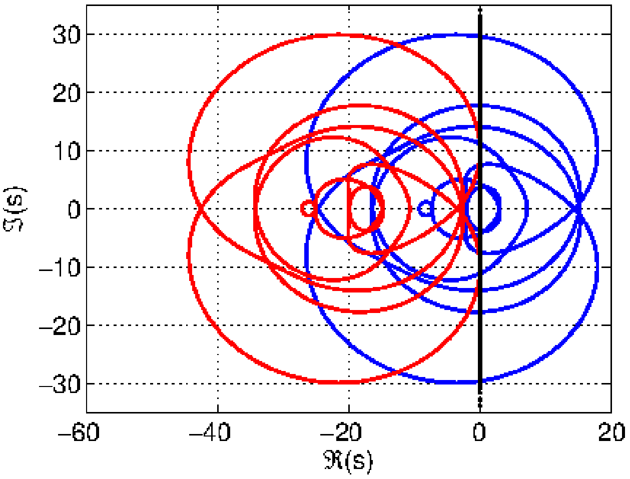


Figure 4.5: Solutions of Eq. (1.17) (in blue) and of Eq. (4.99) with $\bar{l} = \pi$ for Example .4.4.0.1

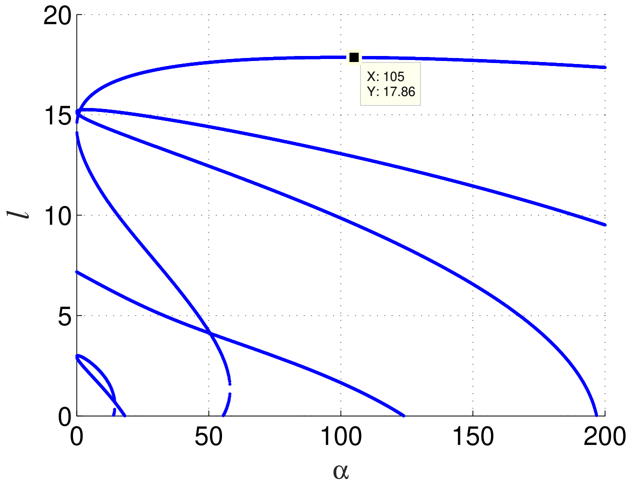


Figure 4.6: Positive real solutions of $R(\bar{\alpha}, l)$. The value of $\bar{l}$ is the maximum, i.e. $\bar{l} =$ 17.87.

as the maximum positive real solution of the remainder $R(\alpha, l)$. Using this value, the closed matrix $\bar{\mathrm{A}}_0$ leads to a DISS as clearly shown in the red curves in Figure. 4.5.

In this section, three case studies are considered to illustrate the effectiveness of the presented method. In the first case, only one delay parameter is considered with a fifth-order system. The second case takes into account three delay parameters with a third-order system and it shows that the method can handle many delay parameters. In the third case, by considering a DDSS, the proposed procedure is used to find the gain parameter of the controller which makes the systems a DISS.

4.5 TWO VARIABLES TO STABILIZE THE DELAY DEPENDENT SYSTEM: STABILITY CHART

4.5.1 Proposed Approach

After designing a proportional controller gain that allows to make the system delay independent stability, a system dependent on the controller gain and an other variable to make a *2-D* stability analysis are designed.

The two variables will be used to construct a stability intervals of a two dimensional (2D) map, known as stability chart, in which the intervals are displayed with respect to a controller gain.

Let us consider a CMTDS as in Eq. (4.116) with A_0 and $\mathrm{A}_0 + \sum_{k=1}^{m} \mathrm{A}_k$ stable matrices. It exists a linear state-space regulator with gain matrix L_0 and a variable β for which the system is DIS such that the controlled system, expressed as

$$\dot{\mathrm{x}}(t) = \frac{\bar{\mathrm{A}}_0}{1+\beta}\mathrm{x}(t) + \sum_{k=1}^{m} \frac{\mathrm{A}_k}{1+\beta}\mathrm{x}(t-k\tau) \tag{4.116}$$

where $\bar{\mathrm{A}}_0 = \mathrm{A}_0 - \mathrm{L}_0$ and $\mathrm{L}_0 = l\mathrm{I}$, with l being the scalar controller parameter gain and I is the identity matrix.

The proof presented in Section. 4.2 could be followed to demonstrate the existing of the two variables l and β.

4.5.2 Algorithm to Plot the Stability Chart

The characteristic equation of the controlled system is $B(s,l,\beta)$ as in Eq. (2.38), which is a function of s, l, and β. The condition in which the characteristic equation admits a pair of purely imaginary solutions is found by varying l and β. Due to the continuity of the roots of $B(s,l,\beta)$ with respect to l and β, this is the limit condition for roots passing from the RHP to the LHP. The condition can be retrieved performing the polynomial division between $B(s,l,\beta)$ and a generic polynomial $B_1(s) = s^2+\alpha$, with $\alpha > 0$, and searching the values of α, l, β for which the remainder is null, i.e. $R(\alpha,l) = 0$.

The algorithm is based on fixing different values of α, β and calculating the roots of $R(\alpha,l)$ with respect to l. The value $\bar{l}$ is chosen as $\max_l \{R(\alpha,l) = 0, \forall\alpha\}$.

Ex 4.5.2.1 In this section, a case study is presented to show the stability chart for l and $\beta withthefollowingsystemmatrixwithn = 3andk = 1:$

$$\mathrm{A}_0 = \begin{pmatrix} -1 & 13.5 & -1 \\ -3 & -1 & -2 \\ -2 & -1 & -4 \end{pmatrix} \tag{4.117}$$

$$\mathrm{A}_1 = \begin{pmatrix} -5.9 & 7.1 & -70.3 \\ 2 & -1 & 5 \\ 2 & 0 & 6 \end{pmatrix} \tag{4.118}$$

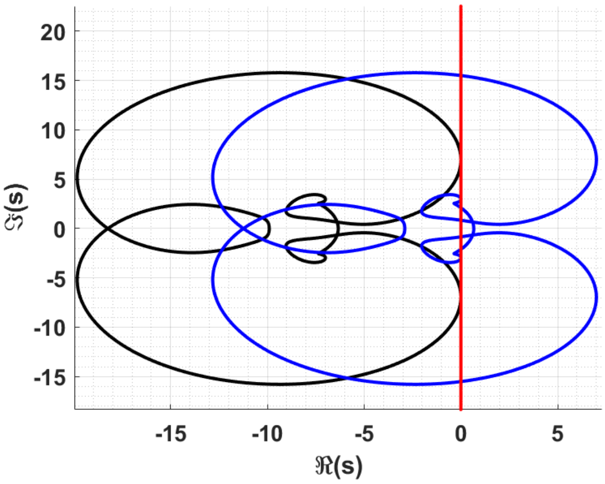


Figure 4.7: Solutions of Eq. (1.17) (in blue) and of Eq. (4.116) with $\bar{l} = 3.6$ *and* $\beta = 0.6$ for Example .4.4.0.1

The correspondent characteristic equation can be written as:

$$a(s,\tau) = (s^3 + 6s^2 + 45.5s + 111) + (0.9s^2 - 116.8s - 22.1)z \\ +(90.9s - 185.1)z^2 + 119.4z^3 \tag{4.119}$$

The stability chart with respect to the controller gain and the variable β following the procedure described above is shown in the plot in Figure. 4.8. This stability chart provides the regions where the system is DDS or DIS.

Based on the results illustrated in Figure. 4.8, the combination of the variables l and β are adopted to have a system DIS. As shown by the roots of the characteristic polynomial in Eq. (1.17) reported in blue in Figure. 4.7 the system is DDS. By using $\bar{l} = 3.6$ and $\beta = 0.6$, the roots of the characteristic polynomial in Equation. (4.116) are obtained in black as reported in Figure. 4.8 and it shows that the system is DIS.

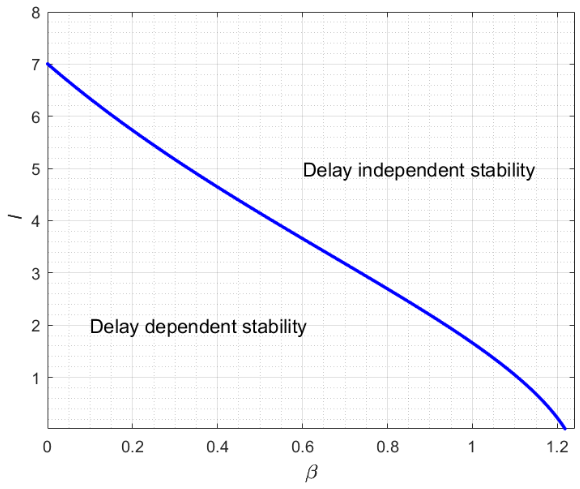


Figure 4.8: Stability chart

4.6 CHAPTER SUMMARY

In this chapter, the design of a state feedback controller is explained and it is theoretically demonstrated that such a controller can transform the system into a DISS.

The proposed controller, which uses only one parameter, can translate the roots of the characteristic equation of the system from the unstable region to the stable region, for any value of the time delay. As a further result, it is shown that the controller gain can be obtained by an easy graphical method, based on the Schur-Cohn criterion and the frequency sweeping method.

As can be observed from the two case studies, the proposed controller design procedure is efficient and simple, even for multiple time delays. Finally, based on the above mentioned assumption, a stability chart using two variables was presented to show the delay dependence stability regions.

CHAPTER 5 MODEL BASED DESIGN APPLICATIONS: A DISCUSSION ON THE ROLE OF TIME DELAY SYSTEMS

5.1 CHAPTER OVERVIEW

In this chapter, the Model-Based Design approach is adopted to model the time delay in a feedback control systems.

The MBD approach is introduced with it is benefits, then an example application is built to understand how the MBD helps engineer to speed their designs especially when time delays are presented within a loop.

Moreover, the chosen of the application was based on the fact that the stability of a PMSM FOC drive is a high sensitive problem and the system can become unstable for different reasons. Time delay is one of the main reasons that can cause the PMSM to become unstable, therefore, for high-speed PMSM motors, it is important to check the stability of current and speed control loops while considering time delays.

This chapter is organized as follows: Section 5.2 gives a brief overview of MBD methodology definitions; Section 5.3 presents how *Simulink* models time delays; Section 5.4 describes the STM32 FOC PMSM Drive application. It presents the hardware and software for modeling the PMSM motor drive; Section 5.5 examines the implementation of the MBD approach for the STM32 motor drive ecosystem; Section 5.6 presents the simulation and code generation results; Finally, in Section 5.7 presents the main source of delays in the PMSM FOC and how the delays affects the stability of such system.

5.2 INTRODUCTION TO MODEL-BASED DESIGN

Many industries and control systems engineers need to reduce time-consume and errors when they start the development of a new product. Working efficiently is indispensable to success in a globalized market [56].

Model-Based Design (MBD) methodologies have been introduced for time-saving, cost-effective, design reuse [57, 58]. These methods and tools may permit to obtain the structural response under relevant loads and improve the structural capacity to withstand extreme loads by a specific design solution.

MBD provides a mathematical and visual approach to develop complex systems. It supports and encourages collaboration by providing a common language for cross-functional teams that work in multiple domains.

MBD is used across a wide-variety of industries and applications, including motion control, signal processing, industrial equipment, aerospace, and auto-motive applications. It centers on the systematic use of models throughout the development process for requirements specification, system architecture modeling, design implementation, simulation, automatic code generation, and verification and validation.

The MBD approach is widely considered to be the most important design flow method as it allows to transform the classical design methods from the lab and handwritten to the desktop.

It uses the same elements as traditional development work flows, but with two key differences:

- many of the time-consuming or error-prone steps in the workflow—for example, code generation are automated;
- a system model is at the heart of development, from requirements capture through design, implementation, and testing.

Figure. 5.1 shows the MBD workflow [60] .

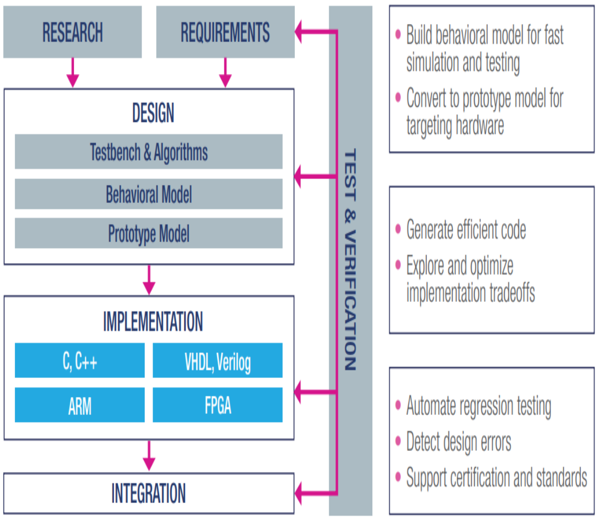

RESEARCH
REQUIREMENTS
DESIGN
Testbench & Algorithms
Behavioral Model
Prototype Model
IMPLEMENTATION
C, C++
VHDL, Verilog
ARM
FPGA
INTEGRATION
TEST & VERIFICATION
• Build behavioral model for fast simulation and testing
• Convert to prototype model for targeting hardware
• Generate efficient code
• Explore and optimize implementation tradeoffs
• Automate regression testing
• Detect design errors
• Support certification and standards

Figure 5.1: MBD workflow

Requirements Capture and Management

In a traditional workflow, where requirements are captured in documents, handoff can lead to errors and delay. Often, the engineers creating the design documents or requirements are different from those who design the system. In MBD, you author, analyze, and manage requirements within your Simulink model. It can create rich text requirements with custom attributes and link them to designs, code, and tests.

Design

In a traditional approach, every design idea must be coded and tested on a physical prototype. As a result, only a limited number of design ideas and scenarios can be explored because each test iteration adds to the project development time and cost. In MBD, the number of ideas that can be explored is virtually limitless. Requirements, system components, IP, and test scenarios are all captured in the model, and because the model can be simulated, it can investigate design problems and questions long before building expensive hard-ware.

Code generation

In a traditional workflow, embedded code must be handwritten from system models or from scratch. With MBD, instead of writing thousands of lines of code by hand, it allows to generate code directly from the model, and the model acts as a bridge between the software engineers and the control systems engineers. The generated code can be used for rapid prototyping or production.

Test and Verification

In a traditional development workflow, test and verification typically occur late in the process, making it difficult to identify and correct errors introduced during the design and coding phases. In MBD, test and verification occur throughout the development cycle, starting with modeling requirements and specifications and continuing through design, code generation, and integration.

Simulink [59] from *Mathworks* is a block diagram environment for a multi-domain simulation that can be used to implement Model-Based Design. Beyond *Simulink*, the required toolboxes for the presented use case example are *Simscape* Electrical, State-flow, Embedded Coder.

5.3 MODELING TIME DELAYS IN SIMULINK

In *Simulink*, there exist both Continuous and discrete Time Delays blocks. SubLibrary contains the blocks of continuous library described below (see *Figure 5.2*).

The Transport Delay Block delays the input by a specified amount of time. It can be used to simulate a time delay. At the start of the simulation, the block outputs the Initial input parameter until the simulation time exceeds the Time delay parameter. The Time delay parameter must be non negative. Best accuracy is achieved when the delay is larger than the simulation step size.

The Variable Time Delay Block appear as two blocks in the Simulink block library. However, they are actually the same builtin Simulink block with different settings of a Select delay type parameter. In the Variable Time Delay mode, the block has a data input, a time delay input, and a data output. The block's output at the current time step equals the value of its data input at a previous time which is equal to the current simulation time minus a delay time specified by the block's time delay input.

The Variable Transport Delay Block The output is equal to the value of its data input at an earlier time which is equal to the current time minus a transportation delay.if we let *u(t)* be the input, $t_d(t)$ the transportation delay, and $y(t)$ the output, then $y = u(t - t_d(t))$

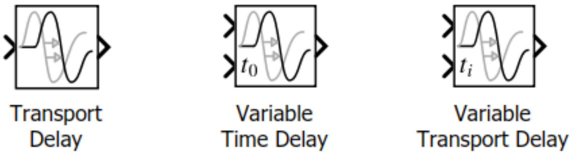


Figure 5.2: Continous time delay in Simulink

Delay discrete-time the Delay block delays a discrete-time input by the number of samples or frames specified in the Delay units and Delay parameters. The Delay value must be an integer value greater than or equal to zero.

In the PMSM DRIVE application, the delay discrete-time will be used because the *Simulink* model is based on discrete time for embedded applications.

5.4 MBD WITH TIME DELAY: STM32 FOC OF PMSM DRIVE APPLICATION

In this section, the STM32 ecosystem both as regard the hardware and the software is described. The target drive system is a PMSM motor with encoder position feedback connected to a 3-phase ac power inverter with three shunt resistors for phase current feedbacks. The drive control algorithm is implemented on a STM32 MCU that includes peripherals dedicated to advanced motor control techniques as shown in Figure. 5.3.

The drive system hardware consists of an X-Nucleo-IHM07M1 power board, a NucleoF302R8 control board, and a 3-PH PMSM motor with encoder feedback. The control board is based on a MCU of the STM32F302x6/8 family. These MCUs are perfectly suited for Motor Control application

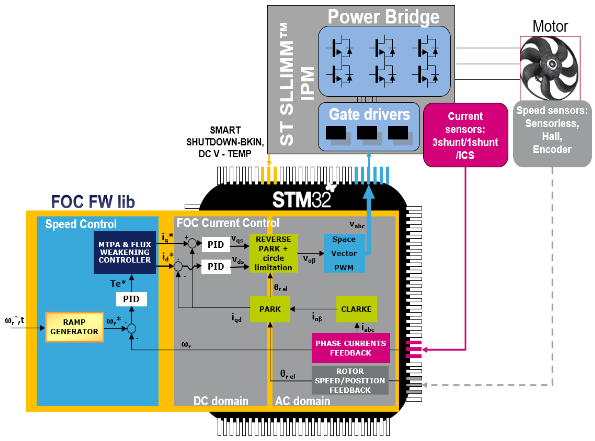


Figure 5.3: Drive system scheme

as they feature a fast 12-bit ADC (5 Msps), three comparators, one operational amplifier, one advanced-timer dedicated to motor control. The STM32F302 MCU executes the drive control algorithm [61] based on the commonly used Field Oriented Control (FOC), whose basic scheme is shown in Figure. 5.4.

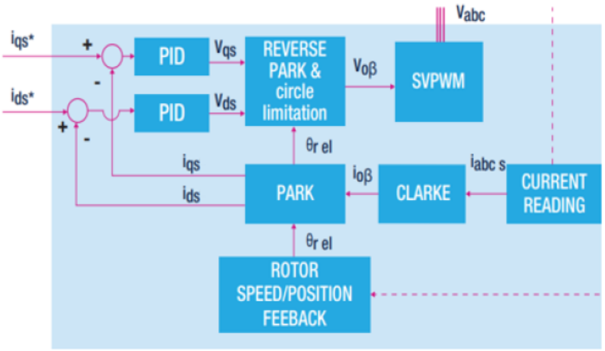


Figure 5.4: FOC scheme

Motor currents and rotor position feedback signal acquisitions are synchronized to the PWM switching frequency using the processor interrupt mechanism. When a new acquisition is completed, the FOC control algorithm executes. In FOC algorithm, sensed motor currents are math transformation performed by Clarke and Park blocks to get two time-invariant components i_{ds} and i_{qs} respectively in phase and quadrature with rotor flux. In this way, it is possible to offer electromagnetic torque T_e regulation by acting on component i_{qs} and using a PID regulator and, to some extent, to operate a flux weakening, by acting on component i_{ds} and using a second PID.

For modeling purposes, the system has three main components: the power inverter and motor (plant), the device interfaces for the control and feedback signals and the digital control unit.

The plant model uses *Simulink Simscape* components to simulate the power inverter electrical circuit and the motor electro-mechanical elements in the continuous-time domain. The feedback circuit models take care of the gain and data type conversions between the controller and motor drive models. The models for the embedded signal interfaces replicate the MCU peripherals by function emulation. They include conversion functions because the ADC converter, the MC dedicated PWM Timer, the encoder timer have 16-bit or 32-bit fixed-point output data registers.

The motor drive system performs several functions beyond the ones dedicated to the motor control algorithm. The embedded software architecture is modular to match both usual demanding requirements for platform flexibility and to make easier the development. These modules typically are devoted to system initialization, communications interface, application tasks, motor control interface, and motor control algorithm as illustrated in Figure. 5.5.

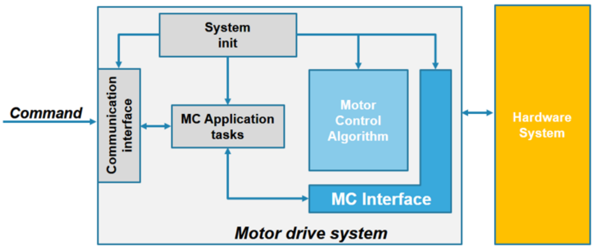


Figure 5.5: Motor drive system tasks

The MC interface module manages the signal data flow between the motor drive hardware and the control algorithm. The code is specific to the drive circuit connections and the MC peripheral configurations needed to provide the appropriate feedback signals for the control algorithm [62].

The system model is partitioned into the logical blocks [63] shown in Table 5.1. Each block is divided into sub-blocks.

Table 5.1: Model partitioning

Block	Modeling/Code Generation
Electromechanical System	Inverter/Motor/Mech-System
Sensors and Interfaces	Function peripherals models
Processor	Peripherals/Code algorithm
Driver Circuits	Function models

In the next section, two dedicated Simulink toolboxes, respectively dedicated to Motor Control Algorithm and STM32 peripheral driver will be described and used to implement MBD workflow.

5.5 IMPLEMENTATION DETAILS

A PMSM Motor drive model has been realized to show the application of MBD concepts to design an embedded control system. This model is used for both drive simulation and code generation for target device. In order to exploit all the potentialities offered by MBD approach, the designer should be given the chance to continue using existing application-dedicated tools offered by ST when start using new tools of *MATLAB/Simulink* environment. For this reason, the ST Motor Control Workbench (STMCWB) tool is used to generate a system configuration file (*.ioc* file) and it is passed as an input to the *Simulink* environment through *MC_Config* block to set model parameters.

Maths and IPs blocks dedicated to FOC algorithm has been strictly built on STM32 Motor Control library and they allow accurate simulations (in normal, SIL and PIL modes) replicating the same behavior of the application running in the real STM32 hardware system. The application designer can configure all the system parameters using the most appropriate method, conceive his project completely in the desktop, exploit all the powerful toolboxes available from *Mathworks* to test and validate the results, analyze wave-forms, and so on. Finally, numerical approximations generally introduced by micro-controller based calculations (due to fixed-point mathematics, for instance) are also taken in consideration at this stage.

Once the simulation results match the project requirements, the second important aspect of MBD methodology is the automatic code generation for target device. By leveraging on STM32 Motor Control SDK and the Embedded Coder tool from Simulink, the application model will generate embedded code (in C or C++ language) in order to create the Application Software layer represented by *MC_App* in Figure. 5.6. The software project can be organized into several C modules for code readability and easy debugging.

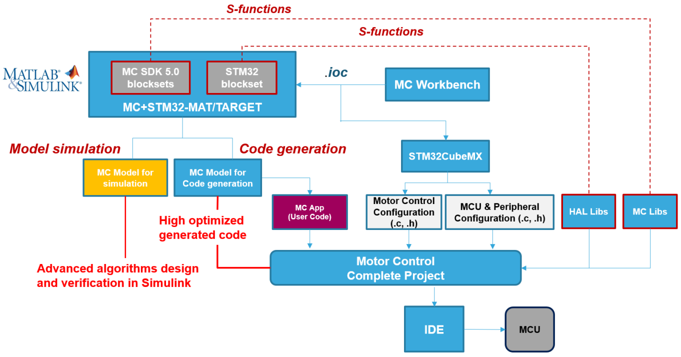

S-functions
S-functions
MATLAB & SIMULINK
MC SDK 5.0 blocksets
STM32 blockset
MC+STM32-MAT/TARGET
.ioc
MC Workbench
Model simulation
Code generation
STM32CubeMX
MC Model for simulation
MC Model for Code generation
MC App (User Code)
Motor Control Configuration (.c, .h)
MCU & Peripheral Configuration (.c, .h)
HAL Libs
MC Libs
High optimized generated code
Advanced algorithms design and verification in Simulink
Motor Control Complete Project
IDE
MCU

Figure 5.6: Modeling and code generation scheme

A detailed explanation of the STM32 MC toolbox is given below.

5.5.1 Simulink Toolbox for STM32 Motor Control

A Simulink toolbox, STM32-MAT/MC, has been developed for providing a set of motor control libraries that supports maths, algorithms, interface devices, hardware IPs for FOC motor control design as seen in Figure. 5.7.

- STM32 FOC Blocks: It provides *Simulink* blocks for mathematical transformations and a PID controller normally used to design FOC Motor Control algorithm. Each block is an S-Function and it

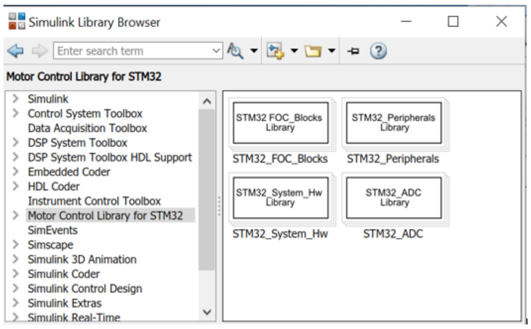


Figure 5.7: Motor control library for STM32

has been coded in C-MEX language by using same function prototypes of C libraries of STM32 Motor Control SDK [62]. A block target file (TLC) has been developed for each block to inline the S-function and to perform an automatic code generation. The Figure. 5.8 shows the mathematical blocks currently available in the *STM32_FOC_blocks* library:

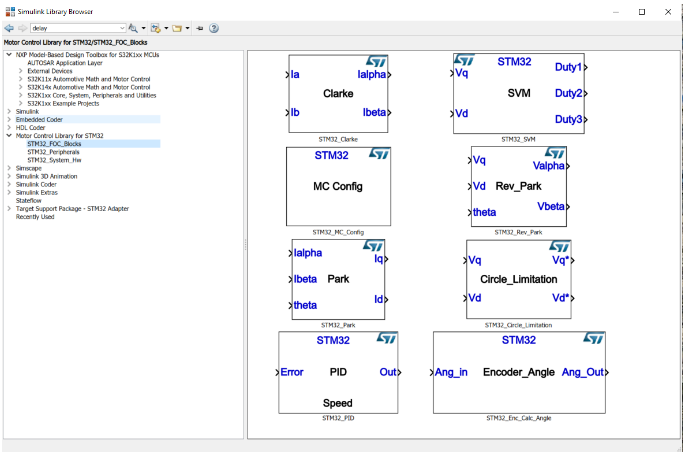


Figure 5.8: STM32 FOC blocks library

- STM32 Peripherals: The *STM32_Peripherals_blocks* library as seen in Figure. 5.9 provides three blocks replicating basic operations of STM32 peripherals: an Advanced PWM Timer, an ADC converter and an Encoder Timer. The functional peripherals were implemented in order to behave, in simulation, like the real ones do. The Timer block emulates the operations of an STM32 Advanced PWM timer by providing three complementary PWM outputs at a specified frequency. The ADC block emulates the operations of an STM32 ADC 12-bit converter which converts input signals under external trigger event occurrence; when a conversion cycle is completed, an external event is generated to run the FOC algorithm. The Encoder timer emulates the operations of an STM32 Timer configured in Encoder mode.

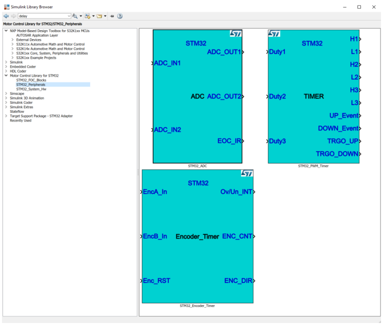


Figure 5.9: STM32 peripherals library

- STM32 System Hw: The *STM32_System_HW* library as seen in Figure. 5.10 provides two basic blocks to model a 3-PH inverter with shunt resistors and an incremental encoder. They can be considered just as off-the-shelf utility blocks to allow the MC designer to use the same hardware devices found in the majority of the motor control kit available from ST.

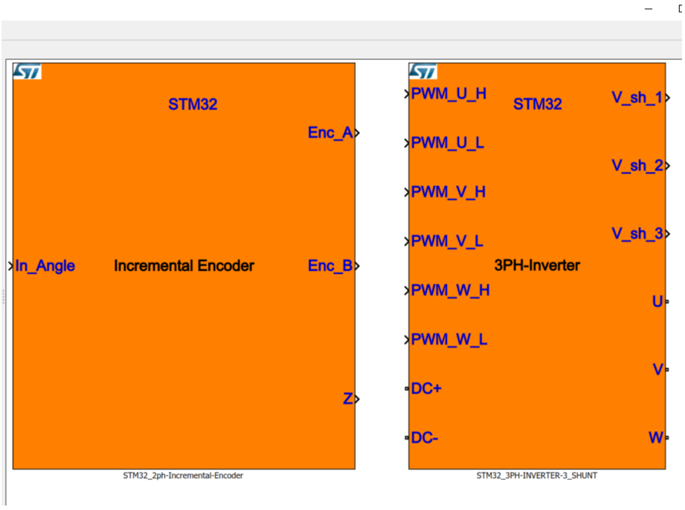
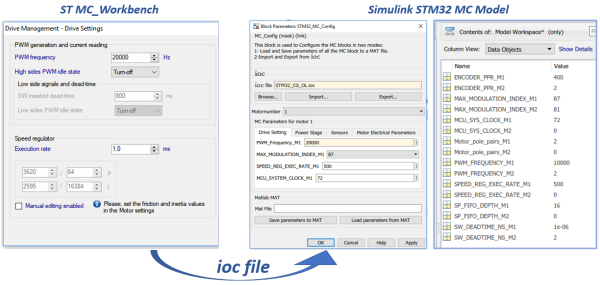


Figure 5.10: STM32 HW library

5.5.2 Model Configuration Parameters

STM32 STMCWB software GUI contains several sections to configure motor drive operations and parameters: PMSM Motor, Power Stage, Drive Management and Control stage just to cite some. Outputs from this tool lead to H files generation that can be included in the MC project. To import configuration parameters set by means STMCWB, a *MC_Config* block is available from STM32-MAT/MC toolbox; it allows configuring each block of model for simulation in a rapid and seamless manner.



Drive Management - Drive Settings

PWM generation and current reading
PWM frequency 20000 Hz
High sides PWM idle state Turn-off
Low side signals and dead-time
SW inserted dead-time 800 ns
Low sides PWM idle state Turn-off

Speed regulator
Execution rate 1.0 ms
3620 / 64 P
2595 / 16384 I
□ Manual editing enabled
Please, set the friction and inertia values in the Motor settings

Block Parameters: STM32_MC_Config

MC_Config (mask) (link)

This block is used to Configure the MC blocks in two modes:
1- Load and Save parameters of all the MC block to a MAT file.
2-Import and Export from ioc

ioc
ioc file STM32_CG_OL.ioc

Motornumber 1
MC Parameters for motor 1

PWM_Frequency_M1 20000
MAX_MODULATION_INDEX_M1 87
SPEED_REG_EXEC_RATE_M1 500
MCU_SYSTEM_CLOCK_M1 72

Matlab MAT
Mat File

Contents of: Model Workspace* (only)
Column View: Data Objects

Name	Value
ENCODER_PPR_M1	400
ENCODER_PPR_M2	2
MAX_MODULATION_INDEX_M1	87
MAX_MODULATION_INDEX_M2	81
MCU_SYS_CLOCK_M1	72
MCU_SYS_CLOCK_M2	0
Motor_pole_pairs_M1	2
Motor_pole_pairs_M2	0
PWM_FREQUENCY_M1	10000
PWM_FREQUENCY_M2	2
SPEED_REG_EXEC_RATE_M1	500
SPEED_REG_EXEC_RATE_M2	0
SP_FIFO_DEPTH_M1	16
SP_FIFO_DEPTH_M2	0
SW_DEADTIME_NS_M1	1e-06
SW_DEADTIME_NS_M2	2

Figure 5.11: Motor control workbench vs *MC_Config*

5.5.3 FOC Engine Algorithm

The FOC control scheme is shown in Figure. 5.12. It consists of blocks from the *STM32_FOC_blocks* library. Each block contains specific parameters that can be set automatically by referring to *MC_Config* block and by specifying the motor drive ID (integer number).

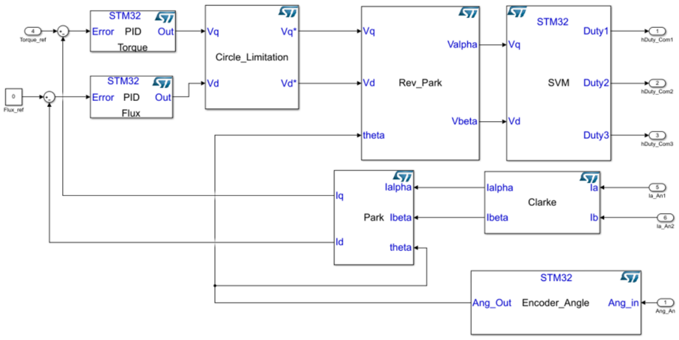


Figure 5.12: FOC engine

5.5.4 Model Architecture

The model architecture presented in Figure. 5.13 follows the model partitioning scheme described in Table 5.1:

This architecture has been conceived to easily move from normal mode simulation, entirely executed on the host PC, to PIL mode simulation, where math's and IPs blocks (and subsystem) are executed in the target STM32 MCU. By using a specific configuration block, this model can be used for both normal simulation and automatic code generation. The only difference between the two configurations consists of the replace-

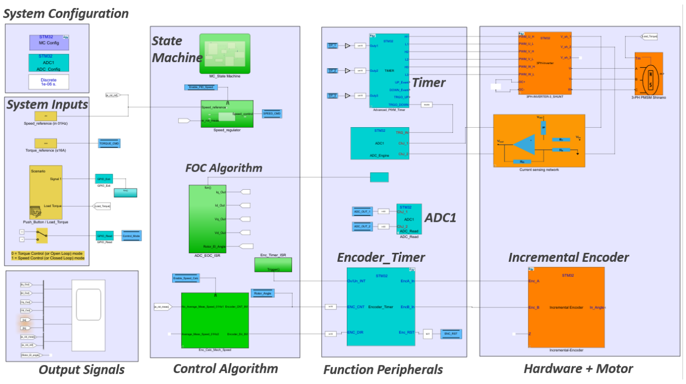


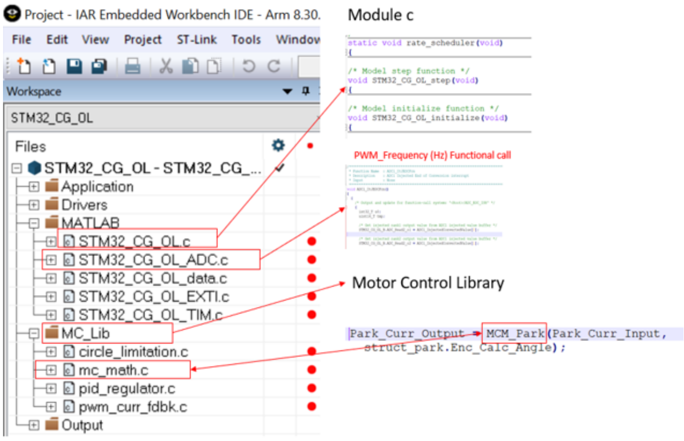
Figure 5.13: Model architecture

ment of functional peripheral blocks into the correspondent ones from STM32-MAT/TARGET [64] peripheral driver library.

5.5.5 Code Generation

As indicated in Figure. 5.14, FOC C code is generated from the motor control algorithm part of the model. The settings for the code generation are in the *Simulink* Configuration parameters → code generation → System Target File, where the target file *stm32.tlc* must be specified. In STM32 Options, the paths for STM32CubeMX and STM32-MAT/TARGET must be specified as well.

The code is organized in a modular fashion that makes integration of application-specific functions easier. High priority tasks, such as the FOC motor control algorithm, are executed in the Interrupt Service Routines at peripheral event occurences. Application-level tasks are called, as scheduled tasks, from a basic scheduler kernel. Simulink Embedded Coder and STM32-MAT/TARGET toolboxes take care of all these aspects automatically.



```
static void rate_scheduler(void)
{

/* Model step function */
void STM32_CG_OL_step(void)
{

/* Model initialize function */
void STM32_CG_OL_initialize(void)
{
```

```
Park_Curr_Output = MCM_Park(Park_Curr_Input,
  struct_park.Enc_Calc_Angle);
```

Figure 5.14: Code module organization

5.6 SYSTEM RESULTS

5.6.1 Simulation Results

By simulating the model shown in Figure. 5.15, relevant signal waveforms can be viewed in the scope Simulink block. For example, Torque Mode was chosen. The scope screen in Figure. 14 has been divided into three subscreens: the first one is the i_q quadrature current, the second one is the motor phase current and the third one represents the three inverter duty cycles computed by SVPWM block. The shapes of waveforms show that the motor runs well.

5.6.2 Code Generation Results

The experimental set-up consists of interfacing the STM32-Nucleo board (connected to the host PC via USB cable) with the inverter board and the power supply and the PMSM motor as shown in section 3. After the project generation from *Simulink* Embedded Coder, the binary application code is built in the user IDE and downloaded in the STM32F302R8

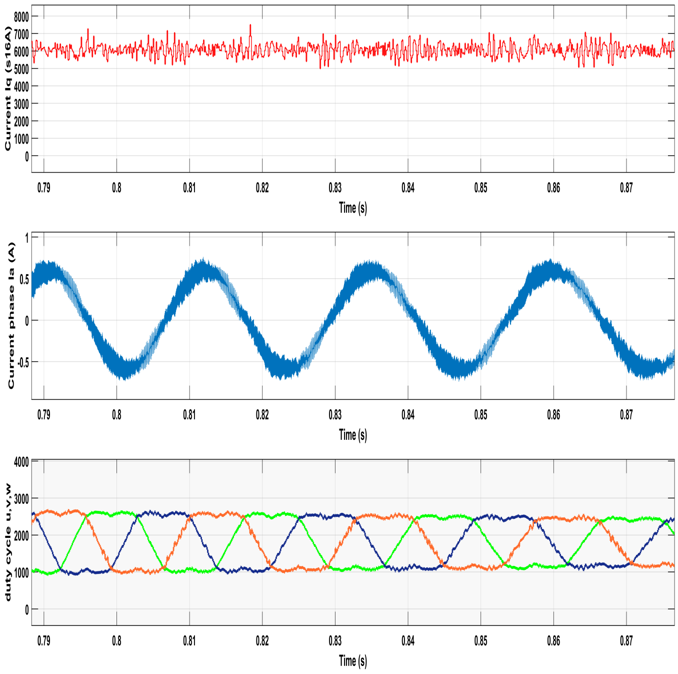


Figure 5.15: Simulation results

MCU. Then, the DC link is supplied with 24V DC. The Figure. 5.16 shows the system. By pushing the user button on the control board the motor shaft starts spinning. Two channels of an oscilloscope were connected to the Nucleo CN10 connector pins to read two PWM outputs. A third scope channel was connected to read the motor phase current.

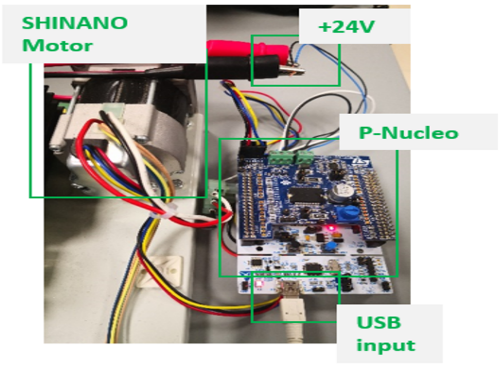
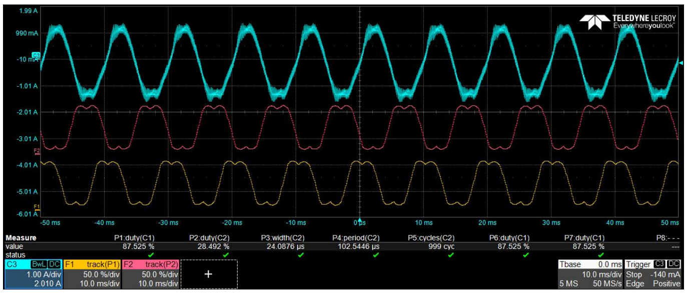


Figure 5.16: The hardware system

The Figure. 5.17 shows a screenshot from the actual oscilloscope: as can be easily seen, the motor waveforms are quite similar to the ones got in simulation. Absolute values for phase current and PWMs are different because torque load conditions, applied to the motor shaft, used in simulation model and in the real system were different.



Measure	P1:duty(C1)	P2:duty(C2)	P3:width(C2)	P4:period(C2)	P5:cycles(C2)	P6:duty(C1)	P7:duty(C1)	P8:- - -
value	87.525 %	28.492 %	24.0876 µs	102.5446 µs	999 cyc	87.525 %	87.525 %	---
status	✓	✓	✓	✓	✓	✓	✓	

C3 BwL DC	F1 track(P1)	F2 track(P2)	Tbase 0.0 ms	Trigger C3 DC
1.00 A/div	50.0 %/div	50.0 %/div	10.0 ms/div	Stop -140 mA
2.010 A	10.0 ms/div	10.0 ms/div	5 MS 50 MS/s	Edge Positive

Figure 5.17: The motor wave-forms, the motor phase current (cyan), the duty cycles (red and yellow)

5.7 SOURCE OF DELAYS

In this section, the source of delays and their block representation on the adopted FOC PMSM control scheme are considered.

Fig. 5.18 depicts a typical speed and current control system for a PMSM FOC motor. In the model different blocks have source of delays, they can be individuated as: FOC Controller (MCU), Actuator (3-phase Inverter), Plant (PMSM motor) and Sensor (the speed and current sensors for signals feedback readings).

Two time delays will be considered: the first affecting the motor current sensing (Sensor block) and the second due to the presence of a digital processing unit device (MCU block). Fig. 5.19 shows the same scheme but considering the insertion of two time delay blocks:

- $e^{-\tau_{delay1}s}$, where τ_{delay1} represents the delay in the current reading path, generated by the current sensing sensor and the analog to digital conversion process necessary for digital processing by the MCU;

- $e^{-\tau_{delay2}s}$, where τ_{delay2} represents the total delay introduced by the FOC algorithm computation time, by the MCU, plus a further time delay due to MCU embedded Timer peripheral generating the PWM signals for the inverter.

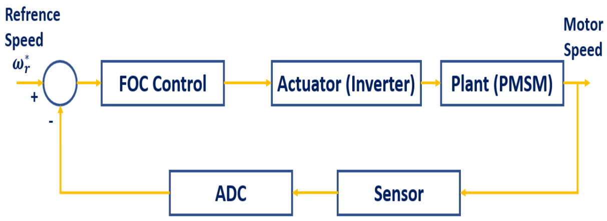


Figure 5.18: Schematic diagram of a speed and current control system controlled by a FOC control

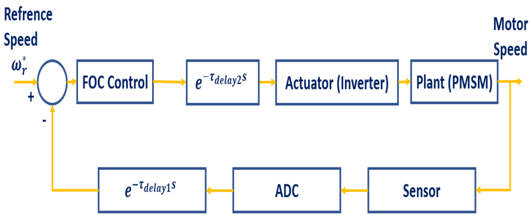


Figure 5.19: Block diagram of Fig.5.18, where the delays are taken into consideration

5.7.1 The Current Sensing Delay Modeling

Now let us take in consideration just the delay due to the current sensing τ_{delay1} as shown in Figure. 5.19. The following practical assumptions are made when modeling the delay of the current sensors.

The currents in the motor constitutes a balanced three-phase current triple. The delay time introduced by the current sensor and ADC converter does not affect the frequency and amplitude of this current triple. The delay time τ_{delay1} is the sum of the current sensors' internal reaction delay (Tsens) and the analog to digital conversion time:

$$\tau_{delay1} = \tau_{DelayCurrSens} + \tau_{DelayADC} \tag{5.120}$$

Where current sensing delay in introduced in 5.121 can be re-written as the formula in 5.122, and it shows that the sensing delay will affect the sensing gain and the phase. 5.122 can be re-arrange as shown in 5.123.

$$i_0 = Asin(\omega t + \varphi_0) \tag{5.121}$$

$$i_{sense} = AKsin(\omega(t - \tau_{delay1}) + \varphi_0) \tag{5.122}$$

where K is the gain due to sensor and ADC.

$$phase = -\omega\tau_{delay1} + \varphi_0 \tag{5.123}$$

5.7.2 The Current Sensing Delay Effects

The simulation was performed for time of 2 s, the delay parameter was increased until the system response become unstable. The delay parameter margin is 6 * T_s i.e. that the system is stable for the value of delay below 300us.

Figure. 5.20 and Figure. 5.21 show respectively *Id* and *Iq* wave-forms for different delay parameter, it can be noticed that by increasing the current sensing delay the signals *Id* and *Iq* starts to be distorted.

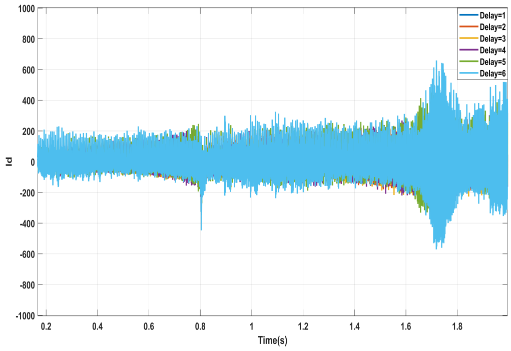


Figure 5.20: Id response with different delay parameter

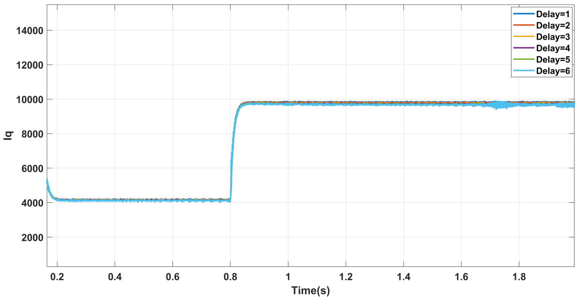


Figure 5.21: Iq response with different delay parameter

This distortion was calculated in term of RMS Error for both signals as illustrated in Figure. 5.22 and Figure. 5.23. It can be seen that the error correspondent to Iq is more higher respect the Id signal.

5.8 CHAPTER SUMMARY

A time delay application example based on Model-Based design methodologies was presented. The implemented MBD approach has been

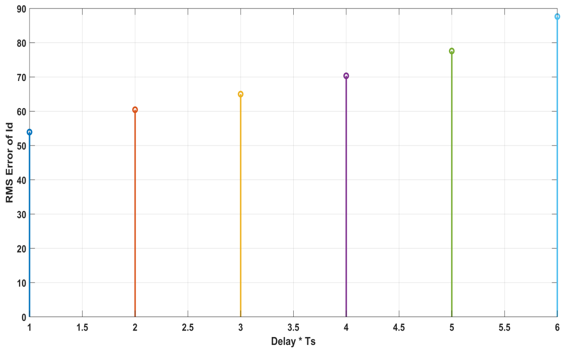


Figure 5.22: RMS Error Id

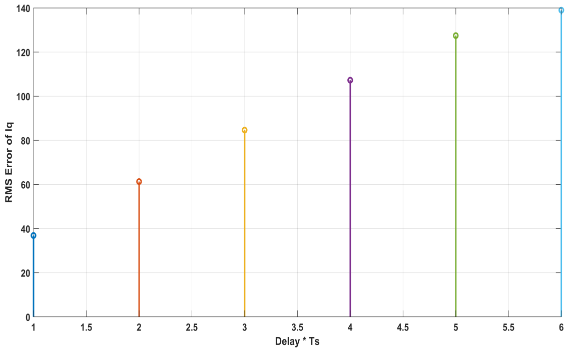


Figure 5.23: RMS Error Iq

conceived as a complementary design path to the traditional one and it provides blocks IP that integrates on existing tools dedicated to the STM32 Motor Control ecosystem. The results shows that increasing the time delay in the feedback loop leads to have an unstable system.

CONCLUSION

Multiple Commensurate Time-Delay Systems appear in various scientific and engineering disciplines. This thesis contributes to the stability analysis of those systems by overcoming the gaps present in state-of-the-art literature.

In the previous chapters, new methodologies were proposed that guarantee the asymptotic stability of linear time-invariant delay systems. Listed below are the main contributions of this research work:

- Provided an approach to obtain the coefficients of time-delay system characteristic equation based on numerical computation;
- Illustrative examples to show how the random number can be chosen to obtain the coefficients of the quasi-polynomial equation;
- Delay-dependent stability criteria to find the imaginary roots of the quasi-polynomial equation of linear commensurate time-delay system;
- Delay-independent stability criteria based on a controller gain able to transform the system from DDS to DIS;
- Stability chart is partitioned into two regions, namely DDS and DIS.
- Design of a STM32 PMSM motor drive model example of application using MBD methodologies also taking into consideration time delays

In Chapter 3, an approach to find the coefficients of the characteristic equation has been developed. The method is based on constructing a matrix system to find the coefficients using suitable random numbers. Then, a graphical method for delay-dependent stability analysis has been presented. It allows determining the imaginary roots in the complex plane. The results of this study indicate that the graphical method is an efficient alternative approach for the stability time delay systems analysis.

In Chapter 4, the control of CMTDS with delay-dependent stability has been addressed. Firslty, the design of a state feedback controller is introduced, that is able to make the system delay-independent stable. The proposed controller, which is based on a single gain parameter, translates the roots of the characteristic equation of the system from the RHP to the LHP, for any value of the time delay. The existence of the stabilizing control gain is theoretically proved in the theorem. As a further result, it is shown that the suitable controller gain can be obtained through both analytical and graphical methods. Three case studies have been discussed to assess the validity of the approach also for multiple time delays. Then, a stability chart based on two variables was illustrated which allows providing the stability regions of commensurate time-delay systems. The chart can handle multiple system delays.

In chapter 5, a time delay application example adopting a Model-based design approach was presented. A new *Simulink* toolbox, STM32-MAT/MC, for STM32 Motor Control design has been presented. It is applied to develop a PMSM motor drive model by taking into consideration time delays. Then, the embedded C code for the STM32F3 target has been automatically generated through the Embedded Coder and the STM32-MAT/TARGET toolboxes. The implemented MBD approach has been conceived as a complementary design path to the traditional one and it provides IP blocks that integrates the existing tools dedicated to the STM32 Motor Control ecosystem. The modeling of time delays was realised in *Simulink* software and given the stable PMSM system, it becomes instable by increasing the time delay in the feedback loop.

The signal waveforms from the simulation model and the real system are comparable, showing that the MBD approach and developed toolboxes are useful tools to speed up and improve the design development cycle. The developed toolbox aims to support ST teams and customers to move from the traditional workflow to a new platform control design approach.

In conclusion, the research in this dissertation will add new theorems and methods to the stability analysis of linear commensurate time-delay systems.

Bibliography

[1] M. Szydliwski, A. Krawiec, The Kaldor-Kalecki model of business cycles as a two-dimensional dynamical system, Journal of Nonlinear Mathematical Physics, 8 (2001), pp. 266-271.

[2] M. R. Roussel, The use of delay differential equations in chemical kinetics, Journal of Physical Chiemistry, 100, pp. 8323-8330. 1996.

[3] F. P. Kelly, Mathema tical modelling of the internet, Mathematics Unlimited 2001 and Beyond, B. Engquist and W. Schmid, Eds. Berlin: Springer-Verlag, pp. 685-702, 2001.

[4] F. Mazenc, S. Mondie and S. I. Niculescu, Global asymptotic stabi-lization for chains of integrators with a delay in the input, in IEEETransactions on Automatic Control, vol. 48, no. 1, pp. 57-63, Jan 2003.

[5] K. M. Sani, A. H. Vette, N. Kawashima, M. R. Popovic, Neuromusculoskeletal torque-generation process has a large destabilizing effect on the control mechanism of quiet standing, Journal of Neurophysiology, 2008.

[6] R. V. Culshaw, S. Ruan, and G. Webb, A mathematical model of cell-to-cell spread of HIV-1 that includes a time delay, J. Math. Biol., vol. 46, pp. 425-444, 2003.

[7] D. Levy, S.-I. Niculescu, P. Kim, and K. Gu, On the stability crossing boundaries of some delay systems modeling immune dynamics

in leukemia, in Proc 17th Int. Symp. Mathematical Theory of Networks and Systems, Kyoto, Japan, pp. 2637-2647, 2006.

[8] J. Miśkiewicz, Economy with the time delay of information flow—The stock market case, in Physica A: Statistical Mechanics and its Applications, vol. 391, pp. 1388-1394, 2012.

[9] Hadeler K.P, Delay equations in biology. In: Peitgen HO., Walther HO. (eds) Functional Differential Equations and Approximation of Fixed Points. Lecture Notes in Mathematics, vol 730. Springer, Berlin, Heidelberg, 2021.

[10] D. Hui, The continuing 2019-ncov epidemic threat of novel coronavirus to global health- the latest: novel coronavirus outbreak in Wuhan, China. Int. J. Infect. Dis.91(2020),pp. 264–266, 2019.

[11] A.Grifoni, D. Weiskopf, Targets of T cell responses to SARS-CoV-2 coronavirus inhumans with COVID-19 disease and unexposed individuals, Cell181, pp. 1–13, 2020.

[12] A. F. Rihan, N. Anwar, M., Sheek-Hussein, M., Denic, S., SIR model of swine influenza epidemic in Abu Dhabi: estimation of vaccination requirement. J. Pub. Health Frontier (PHF)1(4), pp. 85–89, 2012.

[13] A.Seuret, L. Hetel, J. Daafouz, and K. H. Johansson (Eds.), Delays and Networked Control Systems.New York, NY, USA: Springer, 2016.

[14] B.E. F. Camacho and C. Bordons, Model Predictive Control in the Process Industry. London: Springer-Verlag, 1995.

[15] D. M. Prett and M. Morari, Shell Process Control Workshop. Stoneham, MA: Butterworths, 1987.

[16] J. Lagergren, K. Flores, M. Gilman, Deep Learning Approach to the Detection of Scattering Delay in Radar Images. J Stat Theory Pract 15, 14, 2021.

[17] G. Carter, Time delay estimation for passive sonar signal processing,in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 29, no. 3, pp. 463-470, June 1981.

[18] C. Kohrs, N. Angenstein, A. Brechmann, Delays in Human-Computer Interaction and Their Effects on Brain Activity, PLoS ONE 11, 2016.

[19] E. Fridman, Introduction to time delay systems, New York, NY, USA: Springer, 2014.

[20] A.P. Lakshminarayanan, V. Y. Aghav, Ignition Delay in a Diesel Engine. In: Modelling Diesel Combustion. Mechanical Engineering Series. Springer, Dordrecht, 2010.

[21] P. Gharakhanlou, Reducing the time delay of input torque to the electrical motor of EPS system, Amazonia investiga, vol. 8, pp. 479-490, April 2019.

[22] A. Domański, J. Domańska, T. Czachórski, Diffusion Approximation Model of TCP NewReno Congestion Control Mechanism. SN COMPUT. SCI. 1, vol. 43, 2020.

[23] V.V. Kurtc, E.I. Anufriev, Car-following model with explicit reaction-time delay: linear stability analysis of a uniform solution on a ring. Math Models Comput Simul, vol. 9, pp. 679–687, 2017.

[24] S. Hamze, E. Witrant, D. Bresch-Pietri and C. Fauvel, ”Estimating Heat-Transport and time delays in a Heat Exchanger,” 2018 IEEE Conference on Control Technology and Applications (CCTA), pp. 1514-1519, 2018.

[25] M. A. Ardestani, M. Beheshti and M. Najafi, Time delay robust AQM for Internet congestion control,The 3rd International Conference on Control, Instrumentation, and Automation, pp. 6-10, 2013.

[26] X. G. Li, S. I. Niculescu, and A. Cela, Analytic Curve Frequency Sweeping Stability Tests for Systems With Commensurate Delays. New York, NY, USA: Springer, 2015

[27] Kim, J., Joe, H., Yu, S., Lee, J. S., Kim, M, Time-Delay Controller Design for Position Control of Autonomous Underwater Vehicle Under Disturbances. IEEE Transactions on Industrial Electronics, vol. 63(2), pp. 1052–1061, 2016.

[28] R. V. C ulshaw, S. Ruan, and G. Webb, “A mathematical model of cellto-cell spread of HIV-1 that includes a time delay,” J. Math. Biol., vol. 46, pp. 425–444, 2003.

[29] D. Levy, S.-I. Niculescu, P. Kim, and K. Gu, “On the stability crossing boundaries of some delay systems modeling immune dynamics in leukemia,” in Proc 17th Int. Symp. Mathematical Theory of Networks and Systems, Kyoto, Japan, 2006, pp. 2637–2647.

[30] R. J. Anderson and M. W . Spong, Bilateral control of teleoperators with time delay, IEEE Trans. Automat. Contr., vol. 34, pp. 494–501, 1989.

[31] J. E. Speich and J. Rose n, Medical Robotics. New York: Marcel Dekker, 2004.

[32] R. M. Murray, Recent re search in cooperative control of multivehicle systems, ASME J. Dyn. Syst., Meas. Contr., vol. 129, pp. 571–583, 2007.

[33] A. Papachristodoulou and A. Jadbabaie, “Synchronization in oscillator networks with heterogeneous delays, switching topologies and nonlinear dynamics,” in Proc. IEEE Conf. Decision and Control, San Diego, CA, pp. 4307–4312, 2006.

[34] W. B. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs, Proc. IEEE, vol. 94, pp. 1306–1324, 2006

[35] W. Ren and R. W. Beard, Conse nsus seeking in multi-agent systems under dynamically changing interaction topologies, IEEE Trans. Automat. Contr., vol. 50, pp. 655–661, 2004

[36] J. Cheong, S.-I. Niculescu, A. Annaswamy, and M. A. Srinivasan, Synchronization control for physics-based collaborative virtual environments with shared haptics, Adv. Robot., vol. 21, pp. 1001–1029, 2007.

[37] G. Ston, Retarded Dynamical Systems: Stability and Characteristic Function, London, U.K.: Longman Scientific, 1989.

[38] H. Logemann and S. Townley, The effect of small delays in the feedback loop on the stability of neutral systems, Syst. Contr. Lett., vol. 27, pp. 267-274, 1996.

[39] W. Michiel s, K. Engelborghs, D. Roose, and D. Dochain, Sensitivity to infinitesimal delays in neutral equations, SIAM J. Contr. Optim., vol. 40, pp. 1134-1158, 2002.

[40] Delice, I.I. and Sipahi, R., 2011. Delay-independent stability test for systems with multiple time delays. IEEE Transactions on Automatic Control, 57(4), pp.963-972.

[41] X. Li, H. Gao, and K. Gu, Delay-independent stability analysis of linear time-delay systems based on frequency discretization, Automatica, vol.70, pp. 288-294, Aug. 2016.

[42] J. D. Sterman, Business Dynamics: Systems Thinking and Modeling for a Complex World. Boston: McGraw-Hill, 2000.

[43] L. Belhamel, L. Fortuna and M. G. Xibilia, A numerical procedure to obtain the pseudo-polynomial characteristic equation of a commensurate time-delay system, 2019 International Symposium on Signals, Circuits and Systems (ISSCS), 2019, pp. 1-4, 2019.

[44] L. Belhamel; G. M. Xibilia, Graphical Method for the Stability Analysis of Commensurate Multiple Time-Delay Imperfect Systems, IEEE 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC) - Bari, Italy, pp. 1448- 1453 , 2019.

[45] Kamen E., On the relationship between zero criteria for two-variable polynomials and asymptotic stability of delay differential equations. IEEE Transactions on Automatic Control. 1980 Oct;25(5):983-4.

[46] Souza, F.O., De Oliveira, M.C. and Palhares, R.M., 2009. Stability independent of delay using rational functions. Automatica, 45(9), pp.2128-2133.

[47] Gu, K. and Naghnaeian, M., 2010. Stability crossing set for systems with three delays. IEEE Transactions on Automatic Control, 56(1), pp.11-26.

[48] Chen, J., Xu, D. and Shafai, B., 1995. On sufficient conditions for stability independent of delay. IEEE Transactions on Automatic Control, 40(9), pp.1675-1680.

[49] Michiels, W. and Niculescu, S.I., 2007. Stability and stabilization of time-delay systems: an eigenvalue-based approach. Society for Industrial and Applied Mathematics.

[50] Delice, I.I. and Sipahi, R., 2011. Delay-independent stability test for systems with multiple time delays. IEEE Transactions on Automatic Control, 57(4), pp.963-972.

[51] Li, X.G., Niculescu, S.I. and Cela, A., 2015. Analytic curve frequency-sweeping stability tests for systems with commensurate delays. Springer.

[52] Li, X.G., Niculescu, S.I., Cela, A., Zhang, L. and Li, X., 2016. A frequency-sweeping framework for stability analysis of time-delay systems. IEEE Transactions on Automatic Control, 62(8), pp.3701-3716.

[53] Gu, K., Chen, J. and Kharitonov, V.L., 2003. Stability of time-delay systems. Springer Science & Business Media.

[54] Trinh, H., 2015. Refined Jensen-based inequality approach to stability analysis of time-delay systems. IET Control Theory & Applications, 9(14), pp.2188-2194.

[55] Michiels, W. and Niculescu, S.I., 2007. Stability and stabilization of time-delay systems: an eigenvalue-based approach. Society for Industrial and Applied Mathematics.

[56] P. Andriani, F. Conti, L. Fortuna, M. Frasca, G. Passiante, A. Rizzo, Innovation systems by nonlinear networks, Nonlinear dynamics, vol. 44, pp. 263-268, June 2006.

[57] J. Jensen, H. D. C and Chang, A. E. Lee, A model-based design methodology for cyber-physical systems, Proc. 7th Int. Wireless Commun. Mobile Comput. Conf, pp. 1666-1671, July 2007.

[58] J. Reedy, S. Lunzman, Model Based Design Accelerates the Development of Mechanical Locomotive Controls, SAE Technocal Paper 2010-01-1999, 2010.

[59] MathWorks, Simulink® Getting Started Guide (R2021a), www.mathworks.com,2021.

[60] Mathwroks, Model-Based Design for Embedded Control Systems, 2020.

[61] STMicroelectronics, STM32F PMSM single/dual FOC SDK v4.3, User manual UM1052, www.st.com, 2016.

[62] STMicroelectronics, Getting started with STM32 motor control SDK v5.4, User manual UM2374, www.st.com, 2019.

[63] D. O'Sullivan, J. Sorensen, A. Murray, Model-Based Design Streamlines Embedded Motor Control System Development, www.analog.com, 2015.

[64] STMicroelectronics, STM32-MAT/TARGET Hands On Rev 2.4, www.st.com, 2019.

[65] S. Barnett, Polynomial and Linear Control Systems, New York, Marcel Dekker, 1983.

[66] J. Chen, Gu. G, A New Method for Computing Delay Margins for Stability of Linear Delay Sytems, Systems Control Letters 26, pp. 107-117, 1995.

[67] E. K. Walton and J. E. Marshall, Direct Method for TDS Stability Analysis, IEE Proceedings 134, pp. 101-107, 1987.

[68] H. J. Su, The Asymptotic Stability of Linear Autonomous Systems with Commensurate Time Delays, IEEE Transactions on Automatic Control 40, pp. 1114-1117, 1995.

[69] J. Louisell, A Matrix Method for Determining the Imaginary Axis Eigenvalues of a Delay System, IEEE Transactions on Automatic Control 46, pp. 2008-2012, 2001.

[70] V. z. Rekasius, A Stability Test for Systems with Delays, Proc. Joint Automatic Control Conf., Paper No. TP9-A, 1980.

[71] L. Pekař and R. Matuů and P. Dostalek and J. Dolinay, The Nyquist criterion for LTI time-delay systems, 2011.

[72] B. Wayne Bequette, Process Control: Modeling, Design and Simulation, Prentice Hall (2003).

[73] D.E. eborg, T.F. Edgar and D.A. Mellichamp, Process Dynamics and Control, 2nd Ed., John Wiley Sons, New York,2004.

[74] C.A.DESOER, W.S. Chan, Interconnected Linear Multivariable Systems with Delays: System Properties and Input-Output Stability, IEEE Transactions on Automatic Control, , 22, pp. 604-610, 1977.

[75] C.A. DESOER, Y.T. WANG, On the Generalized Nyquist Stability Criterion, IEEE Transactions on Automatic Control, 25, pp. 187-196, 1980.

[76] Y. CHAIT, C.R. MACCLUER, C. j. Radclife, A Nyquist Criterion for Distributed Parameter Systems, IEEE Transactions on Automatic Control, 34, pp. 90-92, 1989.

[77] L. BELHAMEL, A. BUSCARINO, A. CUCUCCIO, L. FORTUNA and G. RASCONA, Model-Based Design Streamlines for STM32 Motor Control Embedded Software System, 2020 7th International Conference on Control, Decision and Information Technologies (CoDIT), pp. 223-228, 2020.

[78] L. Belhamel, A. Buscarino, A. Cucuccio, L. Fortuna and G. Rascona, Model-Based Design for high performance devices oriented to advanced smart solutions, 2020 6th IEEE Congress on Information Science and Technology (CiSt), pp. 442-447, 2020.

[79] C. Dufour, Z. Soghomonian, Li, W, Hardware-in-the-loop testing of modern on-board power systems using digital twins, In Proceedings of the 2018 International Symposium on Power Electronics, Elec-

trical Drives, Automation and Motion (SPEEDAM), Amalfi, Italy, pp. 118–123, 2018.

[80] C.S. Edrington, M. Steurer, J. Langston, T. El-Mezyani, K. Schoder, Role of Power Hardware in the Loop in Modeling and Simulation for Experimentation in Power and Energy Systems. Proc. IEEE 2015, 103, pp. 2401–2409, 2015.