Universidade Federal de Minas Gerais
Programa de Pós-Graduação em Engenharia Elétrica

ANÁLISE ESPECTRAL SINGULAR APLICADA NA FILTRAGEM DE SINAIS EM MALHAS DE CONTROLE

Emerson Alves da Silva

BELO HORIZONTE – MG
OCTOBER 2022

Emerson Alves da Silva

Análise Espectral Singular Aplicada na Filtragem de Sinais em Malhas de Controle

Master dissertation submitted to the Programa de Pós-Graduação em Engenharia Elétrica of the Universidade Federal de Minas Gerais, in partial fulfillment of the requirements for the Master's degree in Electrical Engineering.

Concentration Area: Signals and Systems – Modeling, Analysis and Control of Nonlinear Systems

Advisor: Prof. Dr. Leonardo Amaral Mozelli

Belo Horizonte – MG
October 2022

Silva, Emerson Alves da.
S586a Análise espectral singular aplicada na filtragem de sinais em malhas de controle [recurso eletrônico] / Emerson Alves da Silva. - 2022.
1 recurso online (91 f. : il., color.) : pdf.

Orientador: Leonardo Amaral Mozelli.

Dissertação (mestrado) Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 80-91.

Bibliografia: f. 72-79.

1. Engenharia elétrica - Teses. 2. Análise espectral - Teses. 3. Controladores PID - Teses. I. Mozelli, Leonardo Amaral. II.Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pelo bibliotecário Reginaldo César Vital dos Santos CRB6/ 2165
Biblioteca Profº Mário Werneck, Escola de Engenharia da UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FOLHA DE APROVAÇÃO

"ANÁLISE ESPECTRAL SINGULAR APLICADA NA FILTRAGEM DE SINAIS EM MALHAS DE CONTROLE"

EMERSON ALVES DA SILVA

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica. Aprovada em 13 de outubro de 2022. Por:

Prof. Dr. Leonardo Amaral Mozelli
(UFMG) - Orientador

Prof. Dr. Víctor Costa da Silva Campos - (UFMG)

Prof. Dr. Michel Carlo Rodrigues Leles - (DTECH/UFSJ)

Documento assinado eletronicamente por **Leonardo Amaral Mozelli**, **Professor do Magistério Superior**, em 13/10/2022, às 20:21, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Victor Costa da Silva Campos**, **Professor do Magistério Superior**, em 19/10/2022, às 07:46, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Michel Carlo Rodrigues Leles**, **Usuário Externo**, em 19/10/2022, às 12:00, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1824515** e o código CRC **009E0EEE**.

To my family, for their unwavering support and love.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to God, whose blessings and guidance have been my constant source of strength throughout this journey.

I am profoundly grateful to my advisor, Prof. Dr. Leonardo Amaral Mozelli, for his support, guidance, and patience throughout my research journey. I am also grateful to my committee members, Prof. Dr. Victor Costa da Silva Campos and Prof. Dr. Michel Carlo Rodrigues Leles, for their insightful feedback and encouragement.

Special thanks to my professors, whose knowledge and passion for teaching have greatly contributed to my academic growth. I am deeply appreciative of my laboratory colleagues, whose friendship and collaboration have made this journey enjoyable and fulfilling.

I would also like to extend my sincere thanks to the funding agencies: Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), for their financial support, which made this research possible.

Lastly, I extend my heartfelt thanks to my family, especially my parents, for their endless love and sacrifices. This achievement would not have been possible without you all.

“Trust in the Lord with all your heart
and lean not on your own understanding;
in all your ways submit to him,
and he will make your paths straight.”
(Proverbs 3:5-6)

RESUMO

SILVA, E. A. **Análise Espectral Singular Aplicada na Filtragem de Sinais em Malhas de Controle**. 2022. 91 p. Dissertação (Mestrado Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte – MG, 2022.

Em sistemas de controle linear, controladores PID ainda são amplamente utilizados em controle de processos industriais, onde uma operação eficiente requer o projeto de um filtro. Este filtro viabiliza a implementação de uma ação derivativa e atenua o ruído de medição, reduzindo a variabilidade do sinal de controle. No entanto, ele influencia o desempenho e a robustez do sistema realimentado. Neste contexto, o presente estudo buscou analisar os efeitos da utilização de um filtro não-paramétrico baseado no método Singular Spectrum Analysis (SSA) para tratar ruído de medição em malhas de controle PID e também como filtro de trajetória on-line, necessário em estratégias de controle não linear, tais como *backstepping*, *feedback linearization*. Cujo objetivo é gerar trajetórias contínuas e limitadas derivando-se um sinal inicialmente descontínuo, tais como degrau e pulso. O SSA é utilizado na análise de séries temporais e se baseia na decomposição de um sinal em um conjunto de componentes aditivos, além de ser altamente adaptável ao comportamento dos sinais, inclusive não estacionários. Os testes consideraram modelos First-Order Time Delay (FOTD) para aproximar a dinâmica de processos industriais típicos, para controladores PI e PID sintonizados de maneira ótima, na presença e ausência de ruído de medição. O Integral Absolute Error (IAE) foi utilizado para mensurar o desempenho e a Total Variance (TV) para a variabilidade do sinal de controle e da variável de processo. Como resultados, pode-se destacar que processos dominados por atrasos de transferência apresentaram alta sensibilidade à mudanças no nível de atenuação do filtro em relação aos demais. Em contraste, processos balanceados e com tempo-morto dominante o filtro foi capaz de melhorar tanto a variação total do sinal de controle quanto o desempenho. Além disso, o PID apresentou melhor desempenho do que o PI em todos os processos e cenários considerados, porém para valores maiores de TV devido a ação derivativa. Quanto aos filtros de trajetória, o método SSA se mostrou efetivo como um filtro de trajetória capaz de gerar derivadas limitadas de sinais de entrada inicialmente descontínuos. Cujo ajuste empírico dos parâmetros do filtro levou a curvas semelhantes às obtidas por um filtro paramétrico.

Palavras-chave: Controle de Processos; Filtragem de Ruído; Filtro Não-paramétrico; Ruído de Medição; Análise Espectral Singular.

ABSTRACT

SILVA, E. A. **Análise Espectral Singular Aplicada na Filtragem de Sinais em Malhas de Controle**. 2022. 91 p. Dissertação (Mestrado Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte – MG, 2022.

In linear control systems, PID controllers are still widely used in industrial process control, where efficient operation requires the design of a filter. This filter enables the implementation of a derivative action and attenuates measurement noise, reducing the variability of the control signal. However, it influences the performance and robustness of the feedback system. In this context, the present study sought to analyze the effects of using a non-parametric filter based on the Singular Spectrum Analysis (SSA) method to handle measurement noise in PID control loops and also as an online trajectory filter, necessary in nonlinear control strategies such as *backstepping* and *feedback linearization*, with the objective of generating continuous and limited trajectories by deriving an initially discontinuous signal, such as step and pulse. SSA is used in time series analysis and is based on decomposing a signal into a set of additive components, in addition to being highly adaptable to the behavior of signals, including non-stationary ones. The tests considered First-Order Time Delay (FOTD) models to approximate the dynamics of typical industrial processes, for PI and PID controllers optimally tuned, in the presence and absence of measurement noise. The Integral Absolute Error (IAE) metric was used to measure the performance and the Total Variance (TV) the variability of the control signal and the process variable. As results, it can be highlighted that lag-dominated processes showed high sensitivity to changes in the filter attenuation level compared to the others. In contrast, for balanced and delay-dominated processes, the filter was able to improve both the total variation of the control signal and performance. Additionally, PID showed better performance than PI in all considered processes and scenarios, but for higher TV values due to the derivative action. Regarding trajectory filters, the SSA method proved effective as a trajectory filter capable of generating limited derivatives of initially discontinuous input signals, with empirical adjustment of the filter parameters leading to curves similar to those obtained by a parametric filter.

Keywords: Process Control; Noise Filtering; Non-parametric Filter; Measurement Noise; Singular Spectrum Analysis.

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AMIGO	Approximated-MIGO
APP	Algebraic Pole Placement
CC	Cohen-Coon
CHR	Chien, Hrones e Reswick
CSSA	Causal Singular Spectrum Analysis
D	Derivative
DS	Direct Synthesis
FOTD	First-Order Time Delay
I	Integral
IAE	Integral Absolute Error
IE	Integral Error
IMC	Internal Model Control
ISE	Integral Squared Error
ISTE	Integral Squared Time Error
ITAE	Integral Time Absolute Error
MIGO	M-constrained Integral Gain Optimization
P	Proportional
PD	Proportional-Derivative
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
RSSA	Recursive Singular Spectrum Analysis
SIMC	Skogestad Internal Model Control
SNR	Signal-to-Noise Ratio
SOTD	Second-Order Time Delay
SSA	Singular Spectrum Analysis
SVD	Singular Value Decomposition
TF	Transfer Function
TV	Total Variance
ZN	Ziegler & Nichols

CONTENTS

CHAPTER

1

INTRODUCTION

1.1 Motivation & Objectives

In linear control systems, Proportional-Integral-Derivative (PID) controllers are still the majority of controllers used in the industry, due to their simplicity, few parameters, and cost-effectiveness. According to Åström and Hägglund (2001), they are used in process control, motor drive, optical and magnetic memories, automobiles, flight control, and instrumentation. PID controllers require the design of a low-pass filter to make the derivative action realizable.

This filter is also conveniently used to attenuate high-frequency measurement noise, inherent in sensor devices in closed-loop systems. Given that excessive noise leads to undesirable deviations in the control action, causing excessive actuator motion. As a mechanical device, this variation can lead to the wear of its internal components and malfunction.

On the other hand, the filter dynamics impact the performance and robustness of the control system. Thus, an attempt to make the system more robust to the measurement noise leads to a conflict: higher measurement noise attenuation versus better performance and robustness.

Given this, the first objective of the present study was to investigate the effects of using a real-time non-parametric noise filter based on the Singular Spectrum Analysis (SSA) method in a linear control system with PID controllers. Analyzing, in an exploratory way, the impact of filtering on the closed-loop system, in terms of the performance and variability of the control action and process variable, for different degrees of measurement noise attenuation. For balanced, lag, and delay-dominated process dynamics, PI and PID controllers, and scenarios with and without measurement noise.

In Motion control and industrial robot control problems usually require online computation of smooth and bounded trajectories and their derivatives from a rough input signal. They are also necessary for the design of nonlinear control laws such as feedback linearization,

backstepping, and Active Disturbance Rejection Control (ADRC) techniques. Additionally, abrupt changes in an input signal often lead to vibration, stress, and wear of mechanical components. Therefore, the smoothness of trajectory signals can impact the overall performance of control systems.

A common way to generate smooth and bounded trajectories is using a filter-based approach in which a rough input signal and its derivatives are smoothed by successive applying Finite Impulse Response (FIR) Moving Average (MA) filters. They are simple, efficient, and capable of complying with kinematic constraints imposed on the shape of the trajectories. Another approach is to use a data-driven method, such as the non-parametric SSA filter.

Considering this, the second objective of this study was to investigate the use of a real-time version of the non-parametric SSA filter, known as Causal Singular Spectrum Analysis (CSSA), as a novel online trajectory generating approach. Based on a cascade of CSSA filters that successively smooth an initially rough input signal, such as a step, pulse, and sawtooth signals and its derivatives, to generate smooth and bounded trajectories.

Nonparametric filters differ from the parametric-based approach by not assuming a stationary and known statistical distribution for the signal, e.g., Gaussian. This way they can better deal with model nonlinearities, by adapting the output according to the structure of the data.

The SSA method can adapt to the behavior of the signal. It decomposes the original signal into sub-components representing the time series trend, oscillations, and noise. It does not require statistical assumptions about the original time series or its components to be satisfied. A disadvantage of the approach is the need to tune two parameters to vary the noise attenuation degree and the need to define them empirically by trial and error in each application.

1.2 Outline

This master's thesis is organized as follows. Chapter 2 presents some preliminary concepts and mathematical background, including the PID controller in section 2.1, controller design and measurement noise in section 2.2, the SSA in section 2.4, and the trajectory filter in section 2.3. A brief overview of some PID tuning methods are provided in Appendix A. In Chapter 3, the basic SSA method is described in section 3.1, its real-time version in section 3.2, and its frequency response approach in section 3.4. Chapter 4 presents the results for the SSA applied in PID control loops in section 4.1 and in the trajectory generating scheme in section 4.2. Finally, Chapter 5 presents the conclusions and suggestions for future works.

CHAPTER

2

PRELIMINARY CONCEPTS

This chapter gives some preliminary and mathematical backgrounds for the remaining thesis. It begins with an introduction to the PID controller, followed by some tuning methods commonly used to design PID controllers. Subsequently, the controller design problem with requirements on measurement noise attenuation is described, with some related works. Then, the non-parametric SSA method is introduced, and related works in the context of noise attenuation in time series and control engineering are given. It ends with a description of trajectory filters and some related studies.

2.1 The PID Controller

A PID control law is composed of three terms, as shown in Equation 2.1. The Proportional (P) term, multiplies the error, $e = y_{sp} - y_m$, defined in the feedback loop of Figure 1, the Integral (I) term, proportional to the accumulated error over time, and the Derivative (D) term, proportional to the derivative, or the rate of change of the error. The integral term minimizes the zero steady-state error while the derivative term acts as a prediction on the future values of the error.

$$u(t) = k_c \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \tag{2.1}$$

The controller parameters are the proportional gain k_c, the integral time T_i, and the derivative time T_d.

The Transfer Function (TF) of Equation 2.1 relating the control action $U(s)$ with the error $E(s)$ are shown in Equation 2.2 and 2.3 in their parallel or non-interacting form and

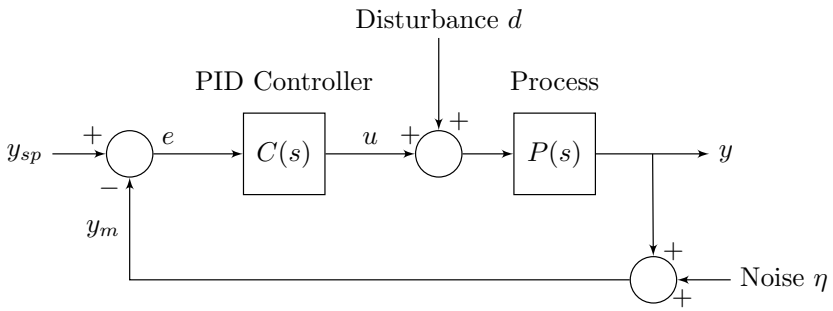


Figure 1 – Block diagram of a feedback loop with an ideal PID controller $C(s)$, plant $P(s)$, set-point y_{sp}, error e, control action u, disturbance d, system output y, and measurement noise η.

illustrated in Figure 2 (a).

$$C^p_{\text{PI}}(s) = \frac{U(s)}{E(s)} = k_c\left(1 + \frac{1}{T_i s}\right) = k_c + \frac{k_i}{s} \tag{2.2}$$

$$C^p_{\text{PID}}(s) = \frac{U(s)}{E(s)} = k_c\left(1 + \frac{1}{T_i s} + T_d s\right) = k_c + \frac{k_i}{s} + k_d s \tag{2.3}$$

where $k_i = k_c/T_i$, and $k_d = k_c T_d$.

Equation 2.4 and 2.5 are the series or the interacting form of Proportional-Integral (PI) and PID algorithms, illustrated in Figure 2 (b), and are commonly used in commercial controllers (ÅSTRÖM; HÄGGLUND, 2006).

$$C^s_{\text{PI}}(s) = k_c\left(1 + \frac{1}{T_i s}\right) \tag{2.4}$$

$$C^s_{\text{PID}}(s) = k_c\left(1 + \frac{1}{T_i s}\right)(1 + T_d s) \tag{2.5}$$

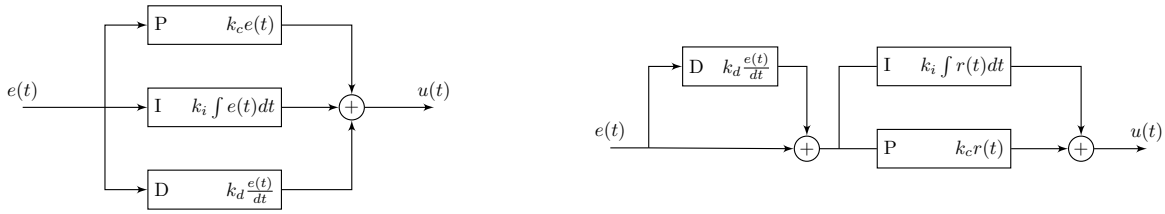


Figure 2 – (a) Parallel and (b) Interacting form of the PID algorithm.

Source: Adapted from Åström and Hägglund (1995).

The practical implementation of the PID requires a filter for the derivative term, also important to limit the controller gain at high frequencies. Equation 2.6 and 2.7 show two common filter choices, a first and a second order low-pass filter.

$$C^f_{\text{PI}}(s) = k_c\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + sT_d/N}\right) \tag{2.6}$$

$$C^f_{\text{PID}}(s) = k_c\left(1 + \frac{1}{T_i s} + T_d s\right)\left(\frac{1}{1 + sT_f + s^2 T_f^2/2}\right) \tag{2.7}$$

where T_d/N and T_f are the filter time constant, with damping $\zeta = \frac{1}{\sqrt{2}}$ and $N = 10$ or 100 (ÅSTRÖM; HÄGGLUND, 2006).

PID controllers have survived many changes in technology, from mechanical and pneumatic controllers to integrated circuits, and from microprocessors to electronic tubes and transistors (YU, 2006; ÅSTRÖM; MURRAY, 2021). According to Bauer *et al.* (2016), the concept of PID control was introduced in the industry between 1930 and 1950, first on a smaller scale, with few pneumatic control loops, until the present day, with large-scale use, with hundreds of PID control loops, implemented digitally.

Bennett (2001) highlights that pneumatic controllers in the mid-1900s provided only proportional action and acted as On-Off controllers, in addition to presenting other characteristics such as high gain and non-linearities. Over time, between 1929 and 1940, there was a need for a controller that performed regulation and not just On-Off control, in addition to the integral and derivative actions, known as reset and pre-act or hyper-reset, respectively (BENNETT, 1993). These names came from the fact that the first aimed at zeroing (or resetting) the systematic error, while the second aimed to anticipate future changes in the error signal.

In 1940 came the need to optimally tune the parameters' gains of the pneumatic PID controllers, which gave rise to the tuning method of Ziegler & Nichols (ZN) (BENNETT, 1996). In the mid-1950s, several process units worked with mechanical, hydraulic, pneumatic, and electrical PID controllers. Which was followed by an increase in the use of electronic components, since they provide all the functions that pneumatic controllers were capable of, in addition to allowing mathematical operations (BENNETT, 2001). Also important is the fact that they reduce the effects of dead-time, which improves performance at low and high frequencies, and their easier connection to reading and recording devices (BENNETT, 2001).

This was followed by the advent of digital computers, which required renewed treatment of aspects such as wind-up of the integral action, sampling, and limited floating-point arithmetic precision. On the other hand, it facilitated the use of the non-interactive version of PID, allowed smoother manual to automatic mode transitions, and handle set-point changes (BENNETT, 2001).

Today's PID controllers are quite different from those used 50 years ago. According to Åström and Hägglund (2006), they are currently used together with logic, function blocks, selectors, and action sequences to perform more complex functions. Allowing its use in more sophisticated control strategies, with the goal of obtaining good regulation, smooth transition, and safe operation, in addition to quick process start-up and shutdown (YU, 2006).

Despite their simplicity, PID controllers are still the most popular way to perform feedback in the industry (WITRANT *et al.*, 2016). Vilanova and Visioli (2012) notes that they

are the most used controllers in the industry due to their excellent cost-benefit trade-off. According to Åström and Hägglund (2006), in process control, they constitute more than 95% of the control loops. Being used in several applications, such as process control, motor drive, optical and magnetic memories, automobiles, flight control, and instrumentation, among others (ÅSTRÖM; HÄGGLUND, 2001).

However, it is a fact that many control loops in industrial process control present low performance due to poorly tuned controllers (HÄGGLUND, 2002; HÄGGLUND, 1999). Ender (1993) points out the fact that approximately 30% of the PI and PID controllers in activity operate in manual mode, underlining that in 65% of control loops, their operation in manual mode was better than in automatic mode due to the poor tuning of the controller's parameters.

Hägglund (1999) cite, as consequences of poorly tuned control loops, the insufficient capacity to reject disturbances, leading to large and long deviations from the set-point. A process that operates under these conditions may impair the quality of the products generated, consume more resources and energy than necessary and result in the need for reprocessing manual intervention, and maintenance.

Based on the historical importance of PID control, its wide use even today in the industry and the operating state of most control loops, studies that seek to improve the operation of such controllers are of fundamental importance.

2.2 Controller Design & Measurement Noise

2.2.1 Controller Design & Measurement Noise Attenuation

A disadvantages of feedback is that measurement noise is introduced into the system (SEGOVIA; HÄGGLUND; ÅSTRÖM, 2013). The main effect of this noise is to cause abrupt changes in the control action (ÅSTRÖM; HÄGGLUND, 2006). These variations cause undesired actuator motion, leading to wear and failure of its mechanical components (SEGOVIA; HÄGGLUND; ÅSTRÖM, 2014a). Measurement noise is generally of high-frequency and is assumed to enter the system additively at the process output (SEGOVIA; HÄGGLUND; ÅSTRÖM, 2014a). To reduce the effects of measurement noise in the closed-loop system, a first or second-order low-pass filter, in this case a parametric one, characterized by the time constant T_f, is traditionally employed (HÄGGLUND, 2012; SEGOVIA; HÄGGLUND; ÅSTRÖM, 2014).

$$G_f^1(s) = \frac{1}{1 + sT_f} \qquad (2.8) \qquad C_f^2(s) = \frac{1}{1 + sT_f + s^2T_f^2/2} \qquad (2.9)$$

However, Segovia, Hägglund and Åström (2013) points out that most controller tuning

methods do not consider the effects of measurement noise during the design phase. The filter time constant is either ignored during the process or chosen after it (HUBA, 2015). In this case, a rule of thumb such as a fraction of the derivative time is employed (GARPINGER; HÄGGLUND; ÅSTRÖM, 2012). Or, as in Segovia, Hägglund and Åström (2014a) and Segovia, Hägglund and Åström (2014b), the inverse of the crossover frequency scaled by a design parameter.

Such empirical rules can have several disadvantages (SEGOVIA; HÄGGLUND; ÅSTRÖM, 2014), such as a controller highly sensitive to measurement noise, leading to excessive control action, or with poor robustness, in the case of a filter with low time constant (ISAKSSON; GRAEBE, 2002). Huba (2015) argues that the derivative filter and its design should be an inherent part of the controller design. Noting that the best way to deal with conflicting goals involved in controller design is to adopt a PID tuning method that considers four parameters, the proportional gain k_c, the integral gain k_i, the derivative gain k_d, and the filter time constant T_f (HUBA, 2015).

However, this filter introduces additional dynamics in the control loop (SEGOVIA; HÄGGLUND; ÅSTRÖM, 2014). According to Segovia, Hägglund and Åström (2014b), the filter changes the system dynamics, and if poorly designed, it can deteriorate robustness and load disturbance rejection. That is one of the reasons why the derivative action is rarely used in industrial process control (ISAKSSON; GRAEBE, 2002; HUBA, 2015; SEKARA; MATAUSEK, 2009).

According to Isaksson and Graebe (2002), the simpler structure of the PI, compared to the PID, which has different configurations (ideal, series, and parallel), is also one of the reasons. In addition, it highlights the lack of tuning methods for PID controllers that also take into account the filter parameters. However, the main reason, according to (ISAKSSON; GRAEBE, 2002; ÅSTRÖM; HÄGGLUND, 2006; KRISTIANSSON; LENNARTSON, 2006; SEKARA; MATAUSEK, 2009), is the fact that the derivative action amplifies the measurement noise, resulting in high variations in the control signal, making a PID more noise sensitive than a PI controller.

Isaksson and Graebe (2002) point out that, although PI can deal satisfactorily with first-order process dynamics, a well-tuned PID controller can achieve significant improvements. As noted in Kristiansson and Lennartson (2006), in which the use of the derivative action significantly improved performance compared to PI, for equal stability margins and only a moderate increase in control activity, for all processes, including those with significant time delay.

In this context, a well-designed filter can even improve the closed-loop performance (HÄGGLUND, 2013). Huba (2015) emphasizes that the success of PID controllers depends heavily on this. This leads to the following compromise: choosing a filter with a given order and structure, seeking to attenuate the measurement noise without impacting too much the

robustness and performance of the system (SEGOVIA; HÄGGLUND; ÅSTRÖM, 2013).

2.2.2 Related Works

This section presents some works that explored the trade-off between robustness, performance, and measurement noise attenuation, using parametric filters combined with different PID tuning methods.

In Segovia, Hägglund and Åström (2013) and Segovia, Hägglund and Åström (2014), the authors investigated the effect that signal filtering has on noise attenuation, control action, load disturbance rejection, and robustness to model uncertainty. They used a second-order Butterworth filter with a single parameter T_f, in series with the controller, to ensure zero gain at high frequencies (i.e., high-frequency roll-off) (ÅSTRÖM; HÄGGLUND, 2006). The evaluation metrics included the maximum gain of TF between the measurement noise and the controller output M_n, in addition to the total variation of the control signal TV_u to characterize noise, the Integral Error (IE) for performance assessment and the maximum value of the sensitivity function M_s for robustness. The PID controllers were designed by the Approximated-MIGO (AMIGO) tuning rules for a maximum sensitivity of $M_s = 1.4$. Processes with balanced, lag-dominated, and delay-dominated dynamics were tested and approximated by a FOTD model. Simple tuning rules were formulated for the filter time constant T_f based on the process and controller characteristics. For PI, T_f is proportional to the normalized time delay of the process $\tau_n = \tau/(\tau + \tau_d)$ and the integral time T_i. For a PID, T_f is proportional to the time derivative T_d. The filter significantly reduced the undesirable control actions caused by the measurement noise, with a moderate deterioration in closed-loop performance and unchanged robustness. The authors also highlighted the importance of extending the rules for more tuning methods and models other than a FOTD.

In the work of Huba (2015), the effects of filtering noise in the closed-loop performance for PI and PID controllers were studied. They used binomial filters with different orders n, characterized by its time constant T_n. The relative total variance was employed to evaluate the control effort, and the IAE metric measured the performance. A modified Multiple Real Dominant Pole (MRDP) method was used to design the controllers and define T_n. A FOTD model approximated the processes in analysis. The simulations considered the cases with and without filtering and the presence and absence of measurement noise. A PI with a filter order $n > 1$ and a PID with a filter of order $n > 2$ showed the best performances. Moreover, a PID with an appropriate filter had superior performance than a filtered PI for a process with lag-dominated dynamics. The derivative action reduced the total variance of the control action and improved the rejection to load disturbances. However, the study did not consider the robustness to model uncertainty.

Sekara and Matausek (2009) proposed a new tuning method based on maximizing the proportional gain instead of the integral gain. It uses four parameters: the proportional

gain k_c, integral gain k_i, the damping of controller zeros ζ and the noise sensitivity M_n, given by the gain of TF relating the measurement noise to the control action, which map to the derivative time k_d and filter time constant T_f. Performance is improved by decreasing the value of ζ in the optimal range from 1 to 1/2 until the minimum value of the IAE, a performance metric related to disturbance rejection. The controller had a first-order low-pass filter with a time constant T_f. The initial values of the controller parameters, used as initial solutions to the optimization problem, were obtained from the M-constrained Integral Gain Optimization (MIGO) method. The results showed that maximizing the proportional gain returns substantial improvements in performance. Even when comparing with similar studies, such as Segovia, Hägglund and Åström (2013) and Segovia, Hägglund and Åström (2014), the proposed method showed better IAE and the sensitivity. The proposed tuning method requires knowledge of the process TF, and the study used only FOTD models for processes with balanced, lag-dominated, and delay-dominated dynamics.

In Kristiansson and Lennartson (2006), the authors studied the optimal and robust tuning of PID controllers for stable and non-oscillatory processes and the trade-off between performance and measurement noise attenuation. The IE, IE $= J_v = 1/k_i$, measured the load disturbance rejection, while the generalized maximum sensitivity GM_s characterized the robustness, given by the maximum weighted gains of the sensitivity and complementary sensitivity functions. And the magnitude of the control sensitivity function measured the control activity J_u, given by the product of the controller's TF with the sensitivity function. The optimization problem was minimizing J_v, or maximizing the integral gain k_i, subject to constraints on the robustness GM_s and on the control activity J_u. A first-order low-pass filter was included in the controller's TF with the filter time constant as a fraction of the process time constant $T_f = \tau/\beta$. Although a different parametrization was used for the controller, the tuning rules were translated to k_c, k_i, and k_d parameters of a PID. They require process characteristics such as the critical process gain k_{crit} and the dominant time constant τ_{63}, i.e., the time it takes for the step response to reach 63% of its steady-state gain. The tuning rules returned competitive results when compared to well-established tuning methods, such as ZN, Internal Model Control (IMC), Skogestad Internal Model Control (SIMC), and methods that assign empirical values to the filter time constant T_f. The proposed tuning method resulted in low control activity, reasonable robustness at high frequencies, acceptable noise sensitivity, good stability margins, and near-optimal performance.

In Segovia, Hägglund and Åström (2014a) and Segovia, Hägglund and Åström (2014b), the authors extended the results obtained in Segovia, Hägglund and Åström (2013) and Segovia, Hägglund and Åström (2014) to the λ-Tuning and SIMC methods. The study addressed the trade-off between robustness, disturbance rejection, and measurement noise attenuation in the design of PI and PID controllers, aiming to design rules for computing the filter time constant T_f from the process characteristics. A second-order Butterworth filter ensured high-frequency roll-off, i.e., limited gain at high frequencies, for PID controllers. As

evaluation metrics, the IAE measured the performance, i.e., load disturbance rejection, the standard deviation of the control signal measured the control activity due to the measurement noise, and the maximum sensitivity M_s, and complementary sensitivity M_t measured robustness. The iterative tuning procedure, the processes dynamics analyzed, and the FOTD approximation models were the same in those studies. The authors observed that filtering substantially reduces undesirable control activities due to measurement noise, with a moderate deterioration in performance, without changes in robustness. It was also possible to formulate rules for choosing T_f based on process parameters such as the dead-time τ_d, the time constant τ, the normalized dead-time $\tau_n = \tau_d/(\tau_d + \tau)$ and α, a trade-off parameter for the λ-Tuning and SIMC methods. For them, a small value of α, 0.05, returned a good compromise between robustness, performance, and measurement noise attenuation.

Micić and Mataušek (2014) proposed a new PID tuning method based on optimization of the proportional gain under constraints on the sensitivity to measurement noise M_n, on the maximum sensitivity M_s, and complementary sensitivity M_p. They considered parallel PID controller structures with set-point weighting and first and second-order low-pass filters. The experimental tests included second and fourth-order Butterworth filters. The processes test batch included stable, oscillatory, integrating, and unstable processes. Optimal values for the proportional gain k_c, k_i, k_d and the filter time constant T_f were obtain by applying the Particle Swarm Optimization (PSO) method with the constraints incorporated in the cost function by penalizing their violation. The results showed fast set-point and load disturbance step response with negligible overshoot, when compared to a PID designed by the SIMC method. The proposed PID based on PSO method showed reasonable robustness to time delays of $\approx 1\,\text{s}$. But they did not seek general tuning rules that considered all these objectives for PID design or consider PI and Proportional-Derivative (PD) controllers in the analysis.

In Šekara, Trifunović and Govedarica (2011) though, the authors explored the trade-off between robustness and performance for higher-order controllers other than PID controllers of second-order, under constraints on robustness and sensitivity to measurement noise. The controller structure was given by the Direct Synthesis (DS) method for a general-order process with time delay. It consists of two adjustable parameters, the time constant λ and the relative damping factor ζ, of the dominant poles of the process. The relative damping ζ controls the trade-off between robustness and performance and influences the sensitivity to measurement noise at high frequencies. The processes tested included 16 representative typical dynamic characteristics, such as delay-dominated, unstable, and integrator dynamics. The higher-order controller showed better load disturbance rejection than optimal PID, with significantly lower IAE and similar robustness for stable processes. Meanwhile, it showed considerably higher robustness and performance for unstable processes than an optimal PID. The proposed method requires adequate knowledge of the process TF and, similarly to the previous studies, did not go as far as deriving general tuning rules based on the process characteristics.

2.3 Trajectory Filter Generators

This section describes two traditional parametric approaches for computing trajectories in control applications. The first is based on a state-space representation, and the other on a cascade of FIR filters. In addition, some related studies on these and more complex trajectory filter approaches are described.

2.3.1 State-Space Trajectory Filter

A parametric trajectory filter in state-space form, as described in Farrell and Polycarpou (2006) and shown in Equation 2.10-2.12, where s is a complex frequency of the Laplace transform.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{a_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \tag{2.10}$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \vdots \\ x^{(n-1)} \\ x^{(n)} \end{bmatrix} = \begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & \dots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \vdots \\ x^{(n-2)} \\ x^{(n-1)} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ a_0 \end{bmatrix} u \tag{2.11}$$

$$y = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \vdots \\ x^{(n-1)} \end{bmatrix} \tag{2.12}$$

In this approach, the low-pass filter in Equation 2.10 is represented in its controllable canonical form in Equation 2.11 and 2.12, which returns, as states, the $n-1$ desired derivatives of the input.

2.3.2 FIR Trajectory Filter

A trajectory filter based on a cascade of FIR filters, as described in Biagiotti and Melchiorri (2012), is depicted in Figure 3. It generates a trajectory with the desired degree of smoothness from a rough input signal, such as steps and pulses. Successive filtering of this input signal f_0 and its derivatives by a MA filter H_i result in a bounded and smooth trajectory.

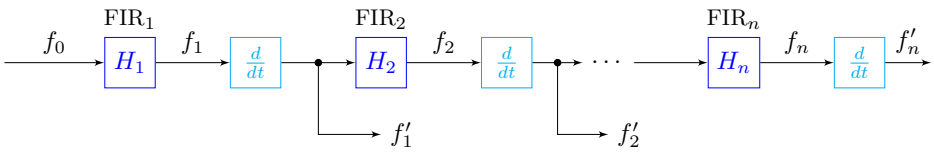


Figure 3 – Cascaded FIR filters to generate trajectories.

The impulse response $h_i(t)$ and the i-th MA filter $H_i(s)$ are given in Equation 2.13 and 2.14, for $i = 1, 2, \ldots, n$, where n is the total number of filters. It performs an averaging operation on the samples of the input signal from $t = 0$ to $t = T_i$. An example of how this process works, for a scaled unit step input $f_0 = au(t)$, for $a > 0$, is presented in Figure 4 and Figure 5.

$$h_i(t) = \begin{cases} \frac{1}{T_i}, & 0 \leqslant t < T \\ 0, & \text{otherwise} \end{cases} \tag{2.13}$$

$$H_i(s) = \frac{1}{T_i}\frac{1 - e^{-sT_i}}{s} \tag{2.14}$$

Assuming, for instance, that $f_0(t)$ represents the position and the velocity trajectory is required. It can be obtained simply by taking the derivative of $f_0(t)$, resulting in $f_0'(t) = a\delta(t)$. As one can see in Figure 4 (b), it is not a bounded signal, has an area equal to a but infinite amplitude. Therefore, it is not suitable for trajectory tracking applications.

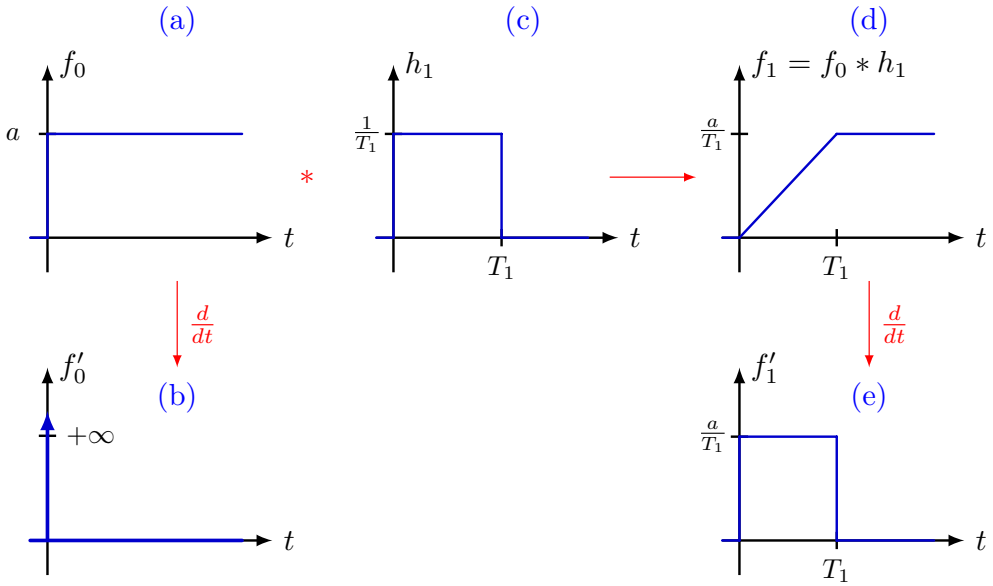


Figure 4 – Effect of filtering a step input signal $f_0(t) = au(t)$ by a MA filter $h_1(t)$.

Instead, before taking the derivative, the input signal $f_0(t)$ is convolved with a MA filter $h_1(t)$, in Figure 4 (c), resulting in a smoother version given by $f_1(t) = f_0(t) * h_1(t)$ in Figure 4 (d), where $*$ represents the convolution operation. Its derivative now, shown in Figure 4 (e), is a bounded signal and can be shaped according to the choice of the filter parameter T_1 and the amplitude a of the input.

Supposing the acceleration profile is also required, to avoid getting an unbounded derivative again, such as $f_1''(t)$ in Figure 5 (f), $f_1'(t)$ is convolved with another MA filter $h_2(t)$, shown in Figure 5 (g), resulting in a signal $f_2(t) = f_1'(t) * h_2(t)$, Figure 5 (i). Its derivative, $f_2'(t)$ in Figure 5 (j), is now a bounded signal. The smoothing effect achieved by successive filtering of the original input signal can be seen by taking the integral of $f_2(t)$, as shown in Figure 5 (h).

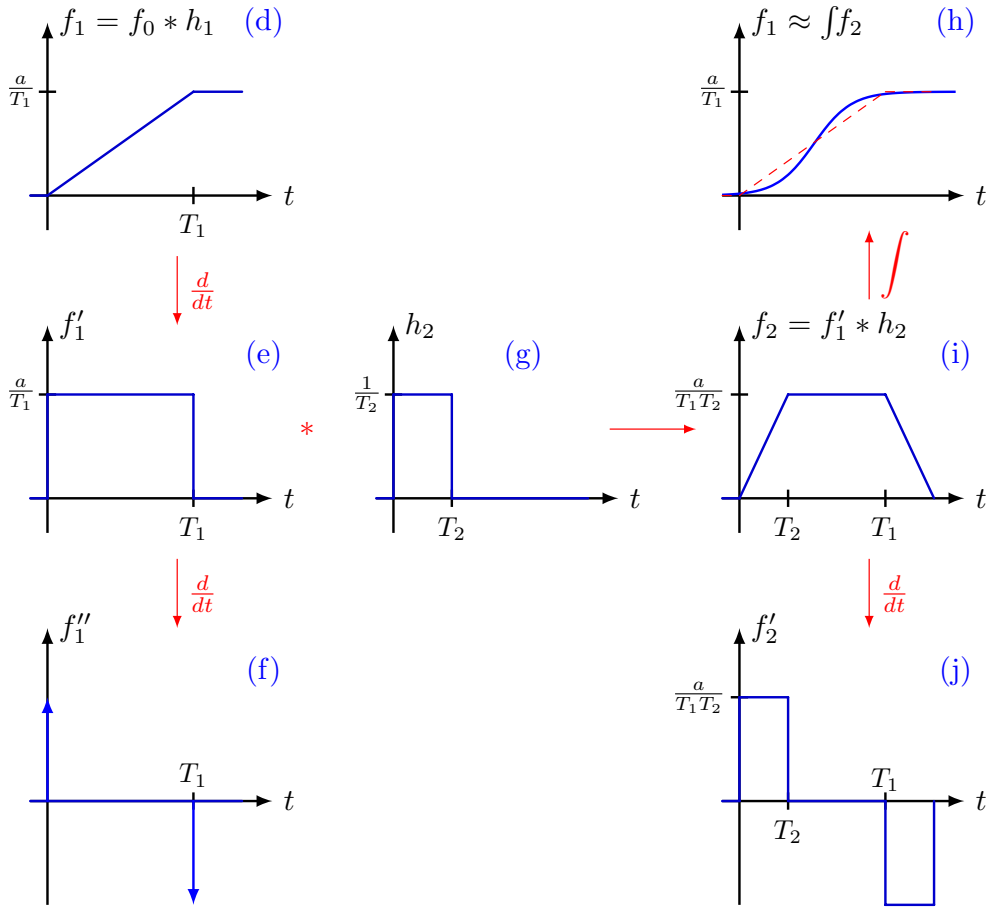


Figure 5 – Further filtering f_1' by a MA filter $h_2(t)$ to get bounded derivative f_2'.

This procedure is then repeated if higher-order derivatives are required. Using a cascade of n FIR filters, the first $n-1$ derivatives are continuous signals, and the n-th derivative is a piece-wise constant signal. Besides, a parametric filter allows restrictions on the minimum and maximum values of the trajectories to be imposed by adjusting the filter parameter T_i, which also controls the degree of smoothness for the filtering.

2.3.3 Related Works

Trajectory schemes aim at producing a set of signal profiles from a rough reference input by computing its derivatives. Usually, the reference input represents the position, and the trajectory scheme generates bounded velocity, acceleration, jerk, snap, and so on, which also satisfy kinematic constraints (BIAGIOTTI; MELCHIORRI, 2012). Gerelli and Bianco (2010) state that controlled robotics and mechatronics applications require smooth signals produced by trajectory generators to improve system performance.

Sudden changes in the control signal can lead to vibration, wear of mechanical components, induced noise, and disturbances (LU, 2008). Zanasi, Bianco and Tonielli (2000), Kim, Jeon and Kim (1994), and Zheng, Su and Müller (2009) emphasize that the

availability of smooth reference trajectories with bounded derivatives is fundamental in getting accurate and quick tracking performance in complex motion control systems. Gerelli and Bianco (2010) stress that smooth reference signals are the most relevant requirement in the control of robotic applications. The bounds on the derivatives, i.e., minimum and maximum velocity and acceleration, are a direct consequence of physical constraints of actuators, such as voltage and current limits (GERELLI; BIANCO, 2010; ZANASI; BIANCO; TONIELLI, 2000).

Several techniques for trajectory generation are available in the literature. In industrial robotic and mechanical systems, smooth trajectories are usually obtained offline by computing optimal profiles (ZANASI; BIANCO; TONIELLI, 2000; ZHENG; SU; MüLLER, 2009). Another offline approach is the selection of polynomial functions (JEON; HA, 2000; ZHENG; SU; MüLLER, 2009). These techniques are not suited for real-time applications due to their high computational cost (ZANASI; BIANCO; TONIELLI, 2000; JEON; HA, 2000). According to Gerelli and Bianco (2010), an online alternative is to generate trajectories by nonlinear filtering through a feedback controller. Studies that explore this approach can be found in Zanasi, Bianco and Tonielli (2000), Zanasi and Morselli (2003), Lu (2008) and Gerelli and Bianco (2010).

Zanasi, Bianco and Tonielli (2000) proposed a nonlinear discrete-time trajectory filter based on a Variable Structured Controller (VSC), sliding mode control with a chain of integrators, capable of providing a minimum-time response without overshoot and symmetrical bounds on the first and second derivatives. Zanasi and Morselli (2003) presented a generalization of this method for higher-order derivatives and a broader class of input signals. In Lu (2008) a time-optimal jerk-constrained trajectory generator with disturbance rejection capabilities was proposed for speed control of electrical drives. Zheng, Su and Müller (2009) used a modified nonlinear tracking differentiator, robust to noise, to generate smooth trajectories from rough input signals such as steps. In Gerelli and Bianco (2010) a VSC, based on an algebraic nonlinear control law, in series with a chain of three integrators, was used to produce continuous trajectories for velocity and acceleration profiles.

Although trajectory filter generators with nonlinear controllers can be successfully used in robotic applications, they are rather complex and computational demanding (BIAGIOTTI; MELCHIORRI, 2012). A simpler online alternative can be obtained through linear filtering with FIR filters (BIAGIOTTI; MELCHIORRI, 2012; BESSET; BÉARÉE, 2017; ZHENG; SU; MüLLER, 2009). Studies that use this filter-based approach are Kim, Jeon and Kim (1994), Olabi *et al.* (2010), Biagiotti and Melchiorri (2012), and Besset and Béarée (2017).

In Kim, Jeon and Kim (1994), a velocity trajectory is generated through convolution with a digital filter for the control of industrial robots and Computer Numerical Control (CNC) machining tools, together with a PID controller and a notch filter for vibration reduction. Olabi *et al.* (2010) proposed a method of jerk-limited trajectory planning for continuous

machining robots that consist of convolving a trapezoidal velocity-shaped signal with a FIR MA filter. Biagiotti and Melchiorri (2012) used a cascade of FIR MA filters, combined with the features of multi-segment polynomial trajectories, to generate time-optimal profiles with kinematic constraints on velocity, acceleration, and jerk. In Besset and Béarée (2017), a low computational complex acceleration-limited signal is convolved with a FIR MA filter to generate jerk-constrained trajectories for robot applications.

2.4 Singular Spectrum Analysis and Related Works

2.4.1 Singular Spectrum Analysis

The SSA is a technique for time series analysis and prediction that includes classic elements of series analysis, statistics, multivariate geometry, dynamic systems, and signal processing (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001; ALEXANDROV, 2009). It decomposes the original time series in terms of its internal components, which are additive and interpretable, such as trend, periodical components, and noise (GOLYANDINA *et al.*, 2013).

This technique can be applied in trend analysis, detection, and extraction of quasi-periodic components, noise attenuation, and point change detection (ALEXANDROV, 2009). According to Hassani (2007), other applications include identification of trends and patterns with different resolutions and complexity, smoothing, extraction of seasonal components, cycles, and periodic signals with varying amplitudes.

It does not require any statistical assumptions to be fulfilled for the time series or its components (GOLYANDINA; NEKRUTKIN; ZHIGLJAVSKY, 2001). It is not necessary also a parametric model that characterizes its statistical distribution, nor non-stationarity conditions (ZHIGLJAVSKY, 2010; HASSANI, 2007). It is simple to use, requires few parameters to be adjusted, and allows trend extraction in the presence of noise and oscillating components (ALEXANDROV, 2009).

The method consists of two complementary stages: decomposition and reconstruction. The decomposition stage is composed of the embedding and the decomposition steps. The reconstruction performs the grouping and the diagonal averaging steps. In the embedding step, the one-dimensional signal is mapped into a multi-dimensional series given by a matrix called trajectory (GOLYANDINA *et al.*, 2013). A key parameter in this step is the window length L. Then, the trajectory matrix is decomposed in terms of its singular values, using the Singular Value Decomposition (SVD) or eigendecomposition, leading to a representation based on a sum of orthogonal elementary matrices of rank one (HASSANI, 2007). The grouping step consists of merging the elementary matrices according to predefined criteria, such as their contribution to the original time series, pattern similarities,

and component correlation. Finally, in the diagonal averaging step, the multi-dimensional groups are mapped back into a one-dimensional time series as additive components of the original series (HASSANI, 2007). In this step, an important parameter is the number R of components selected for the reconstruction.

As Alexandrov (2009) points out, it is similar to the Principal Component Analysis (PCA) technique, which is applied to a multidimensional series, decomposed by the SVD method, and returns a new representation, with a lower dimension and different physical meaning. The SSA, nonetheless, preserves its original dimension and physical meaning throughout the decomposition process. Although relatively new, it is a powerful technique that is being applied to many practical problems in different areas of science (HASSANI; THOMAKOS, 2010).

2.4.2 Related Works

The following studies are examples of works that used SSA in noise filtering and time series smoothing, in areas such as: Bio-mechanics, Alonso, Castillo and Pintado (2005); Geo-dynamics, Chen *et al.* (2013); Materials Science, Salgado and Alonso (2006); Economics and Finance, Hassani, Dionisio and Ghodsi (2010); and in Control Engineering, Haavisto (2010), Zhang and Wang (2016), Phuong *et al.* (2016), Phuong, Ohishi and Yokokura (2019), and Phuong, Ohishi and Yokokura (2020).

In Alonso, Castillo and Pintado (2005), the authors analyzed a stationary sinusoid, a non-stationary bio-mechanical signal, and two others from the literature. They used the SSA and a parametric Butterworth filter to reduce the noise in those time series and allow its numerical differentiation to obtain velocity and acceleration signals. Empirical values were assigned to the parameter window length L. Meanwhile, the number of principal components R was based on each component's eigenvalues, a measure of its contribution to the reconstruction of the original time series. The SSA returned smoother signals, compared to the parametric filter, for both stationary and non-stationary signals. Furthermore, the SSA did not require additional adjusting methods to correct the resulting signal as the Butterworth filter. However, some time series required applying the SSA repeatedly to effectively attenuate noise. A fundamental drawback highlighted by the authors is the lack of rules to assign values to the filter's parameters. They suggest, for future research, the development of automatic parameter selection methods and the successively use of cascaded SSA filters to obtain better results.

In Chen *et al.* (2013), the authors analyze seasonal signals from Global Positioning System (GPS) time series, collected over ten years and referring to weekly monitoring data of deformations, tectonic movements, and displacement of surfaces such as glaciers. The signals' trend can be described as a sum of sinusoids with varying amplitude and phase, mixed with noise. The study used the SSA, Least Square (LS) fit, and the linear Kalman filter

to identify and extract these sinusoidal trends. The window length L was chosen empirically between 2-3 years (105-157 weeks). The SSA and the Kalman filter identified the sinusoids more accurately by correctly capturing changes in the amplitude over time, with SSA showing slightly better results in terms of the mean squared error. In contrast, the LS fit did not identify such variations. By varying the window length L, the SSA extracted almost all the critical information from the original signal, capturing even some nonlinearities.

In Salgado and Alonso (2006), the SSA is used in a tool wear detection analysis. The test stress consists of subjecting a tool to torsion and measuring its longitudinal and transversal deformation with accelerometers. The result of this measurement is a signal embedded in noise, with the trend showing the vibration, the component of interest, also present at high frequencies. The advantage of the SSA method relies on the fact that it is based not only on the frequency of each component but also on the decomposition of the time series in terms of the singular values of the trajectory matrix, allowing the correct identification of the trend. The value of the embedding dimension was $L = 5$, for a time series of length $N = 500$. The high-frequency vibration trend was correctly identified and extracted, without noisy components, with the SSA filter. This signal was then fed into a multi-layer perceptron neural network that relates it to the material wear. The authors also point out that the choice of the main components to reconstruct the time series by the SSA method must be based on a systematic procedure and not just defined empirically.

Hassani, Dionisio and Ghodsi (2010) analyzed a time series of closing prices of stock market indices traded in the US, Portugal, France, Germany, and Greece, containing 4629 observations from 1990 to 2007 for each index. The goal was to assess dependencies in financial time series using statistical correlation metrics robust to nonlinearities. Additionally, the time series are non-stationary and noisy due to abrupt price oscillations in short periods. To smooth the time series and retain the general trend, the authors employed a Autoregressive Moving Average (ARMA) filter, as well as a linear filter, a Generalized Autoregressive Conditional Heteroscedasticity (GARCH), a nonlinear filter, and the SSA. The SSA was able to denoise the time series better than the remaining filters, returning time series with the lowest standard deviation. In contrast, the GARCH method did not identify nonlinearities in the time series. The authors concluded that the quality of the filtered time series significantly improves the correlation indices, thus allowing a more accurate causality analysis.

In Haavisto (2010), a Recursive Singular Spectrum Analysis (RSSA) is used to detect and remove rapid oscillations in a feedback-controlled copper flotation circuit in real-time. Rapid oscillations alter the balance between concentrate grade and valuable mineral recovery, reducing the performance of the mineral processing plant. They applied the SSA due to its time delay and phase-shift invariant characteristics to smoothen the signal, extract the oscillation and detect similarities in the signal structure. The analysis covered 4320 samples of 13 signals from the controlled process. The proposed RSSA method parameters

adopted were $L = 90$ (1.5h) and $\lambda = 0.95$, the forgetting factor. Removing the less significant components/eigenvalues smoothed the signals, highlighting the desired leading sinusoidal oscillations with distinct phases. Moreover, the method allowed the detection of structural changes in the signals after filtering them by computing a heterogeneity metric.

Zhang and Wang (2016) applied an online version of the SSA combined with an Extended State Observer (ESO) and a Positive Position Feedback (PPF) in a ADRC method. To perform attitude control and sloshing suppression in a liquid-filled spacecraft. Sloshing is the motion of a free liquid surface inside a container, which impacts the spacecraft's motion and possibly results in unstable behavior. The ADRC maintains control performance under uncertainties and disturbances. The ESO estimates the total disturbance, and the SSA identifies the sloshing mode from the remaining state variables and extracts it with an Eigensystem Realization Algorithm (ERA) to feed the PPF for damping the sloshing effect. The authors used an embedding dimension of $L = 30$, $N = 60$ samples, sampling interval $T_s = 0.1\,\text{s}$, and reconstructed the original signal with the first $R = 4$ components. The proposed method stabilized the system, while a PD controller failed to achieve. Additionally, the controller showed robustness to changes in the parameters up to $\pm 10\%$ and to low-frequency sinusoidal disturbances.

In Phuong *et al.* (2016) a novel Instantaneous State Observer (ISOB) based on the SSA filter to perform load torque estimation in motion control is proposed. The aim was to improve the performance of the speed control system of an industrial robot arm. Generally, the measurement noise introduced by the accelerometer sensors substantially degrades the system's performance. The experimental results showed reasonable noise attenuation for the proposed method and for the Variable-Noise-Covariance Kalman filter based ISOB, while a 16-point window MA filter failed to reduce vibration (i.e., noise) effectively. In contrast, the SSA-ISOB method showed more precise load torque estimation than the Kalman filter-based method, given the fact that SSA is suitable for non-stationary signals and the Kalman filter assumes a stationary distribution for the noise. They applied an iterative version of the SSA to a batch of experimental data rather than performing the experiments in a real-time system.

Similarly, Phuong, Ohishi and Yokokura (2019) proposed a SSA based Disturbance Observer (DOB) for wide-band sensorless force estimation to improve accuracy of a force-controlled system. The wide-band force estimation is fundamental in the control of interacting robots and is usually highly noisy for this reason. Another undesired component is the harmonic dithering effect caused by a dither signal introduced on the motor side for friction compensation. So, the Harmonic Estimation SSA based DOB method aimed to estimate the force component from the total load torque and the derivative of the position signal while simultaneously suppressing measurement noise and removing harmonic dither signals. The experimental results were compared with the original DOB method, without SSA, with the SSA based DOB achieving more accurate torque estimation with significant noise attenuation

and faster response without time delay. The undesired dither harmonic component was correctly identified and extracted with the SSA and the LS method. Phuong, Ohishi and Yokokura (2020) extended this method to identify and remove undesired periodic components superimposed in the measurements.

CHAPTER

3

METHODS

This chapter describes the basic SSA methodology, followed by its real-time version, the CSSA. Then an example applying the method to a real time series is provided. It proceeds to describe some aspects of the frequency-domain interpretation of the SSA method as a Filter Bank (FB) of FIR filters. An example illustrating this description closes the chapter.

3.1 Basic SSA

The non-parametric SSA filter, in its original form, is applied to a real-valued time series X_N of length N:

$$X_N = \begin{pmatrix} x_0 & x_1 & \cdots & x_n & \cdots & x_{N-2} & x_{N-1} \end{pmatrix} \tag{3.1}$$

The filtering process consists of four steps, divided in two stages. The first is the Decomposition stage, which comprises the Embedding and the SVD steps. The second is the Reconstruction stage, which performs the Grouping and the Diagonal Averaging steps.

1. **Decomposition**

   a) **Embedding**:

   This step maps the original one-dimensional time series in Equation 3.1 into a K-dimensional series $\mathbf{X_m}$, called the trajectory matrix, illustrated in Figure 6.

   The trajectory matrix is made of a set of K-column vectors of length L that are subsets of the original time series, where $K = N - L + 1$. It is a Hankel matrix, given that the elements in its anti-diagonals are the same.

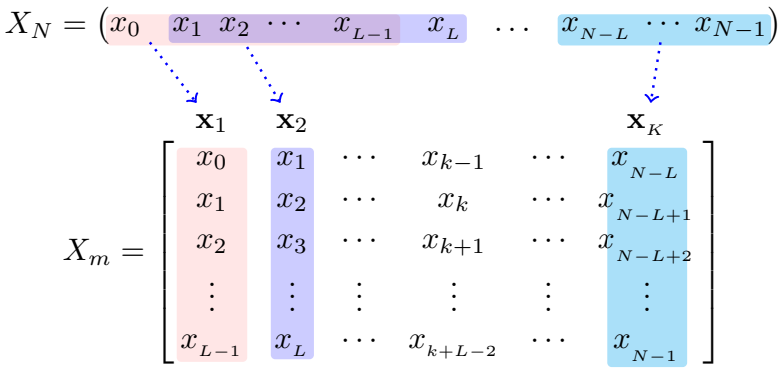


Figure 6 – Trajectory matrix design.

The filter parameter $L \in \mathbb{Z}$ is called the window length or the embedding dimension and is constrained to $2 \leqslant L \leqslant N/2$. It represents the maximum number of components into which the original time series is decomposed.

b) **Singular Value Decomposition (SVD)**:

In the second step the trajectory matrix is decomposed in terms of its right and left singular vectors $\mathbf{U} \in \mathbb{R}^{L \times L}$ and $\mathbf{V} \in \mathbb{R}^{K \times K}$, and singular values $\mathbf{\Sigma} \in \mathbb{R}^{L \times K}$, using the SVD method:

$$\begin{aligned} \mathbf{X_m} &= \mathbf{U\Sigma V}^T \\ &= \begin{bmatrix} u_1^{(1)} & u_1^{(2)} & \cdots & u_1^{(L)} \\ u_2^{(1)} & u_2^{(2)} & \cdots & u_2^{(L)} \\ u_3^{(1)} & u_3^{(2)} & \cdots & u_3^{(L)} \\ \vdots & \vdots & \ddots & \vdots \\ u_L^{(1)} & u_L^{(2)} & \cdots & u_L^{(L)} \end{bmatrix} \begin{bmatrix} \sigma_{11} & 0 & \cdots & 0 \\ 0 & \sigma_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{LK} \end{bmatrix} \begin{bmatrix} v_1^{(1)} & v_1^{(2)} & \cdots & v_1^{(K)} \\ v_2^{(1)} & v_2^{(2)} & \cdots & v_2^{(K)} \\ \vdots & \vdots & \ddots & \vdots \\ v_K^{(1)} & v_K^{(2)} & \cdots & v_K^{(K)} \end{bmatrix}^\top \end{aligned} \tag{3.2}$$

where each column of the singular vectors is sorted in decreasing order of magnitude of its corresponding singular values $\sigma_{11} \geq \sigma_{22} \geq \ldots \geq \sigma_{LK}$.

From the representation in Equation 3.2, the trajectory matrix is then splitted into $q = \max\{i | \sigma_{ii} > 0\} \leqslant L$ elementary matrices:

$$\begin{aligned} \mathbf{X_m} &= \sum_{i=1}^{q \leqslant L} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \\ &= \sigma_1 \underbrace{\begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \\ \vdots \\ u_L^{(1)} \end{bmatrix} \begin{bmatrix} v_1^{(1)} \\ v_2^{(1)} \\ \vdots \\ v_K^{(1)} \end{bmatrix}^\top}_{\mathbf{X_{m_1}}} + \cdots + \sigma_q \underbrace{\begin{bmatrix} u_1^{(q)} \\ u_2^{(q)} \\ \vdots \\ u_L^{(q)} \end{bmatrix} \begin{bmatrix} v_1^{(q)} \\ v_2^{(q)} \\ \vdots \\ v_K^{(q)} \end{bmatrix}^\top}_{\mathbf{X_{m_q}}} \end{aligned} \tag{3.3}$$

Resulting in a set of matrices $X_\mathcal{D}$ that represents the trajectory matrix of each component:

$$X_\mathcal{D} = \{\mathbf{X_{m_1}}, \mathbf{X_{m_2}}, \ldots, \mathbf{X_{m_q}}\}, \quad \mathbf{X_{m_i}} \in \mathbb{R}^{L \times K} \tag{3.4}$$

c) **Eigendecomposition (Alternative)**:

An alternative to the SVD method is the eigendecomposition of the covariance matrix defined in Equation 3.5.

$$\mathbf{C_m} = \frac{\left(\mathbf{X_m}\mathbf{X_m}^\top\right)}{K}, \qquad \mathbf{C_m} \in \mathbb{R}^{L\times L} \tag{3.5}$$

Which results in a representation of the original covariance matrix in terms of its eigenvalues $\mathbf{\Lambda} \in \mathbb{R}^{L\times L}$ and eigenvectors $\mathbf{U} \in \mathbb{R}^{L\times L}$, as shown in:

$$\begin{aligned}\mathbf{C_m} &= \mathbf{U\Lambda U^T} \\ &= \begin{bmatrix} u_1^{(1)} & u_1^{(2)} & \cdots & u_1^{(L)} \\ u_2^{(1)} & u_2^{(2)} & \cdots & u_2^{(L)} \\ u_3^{(1)} & u_3^{(2)} & \cdots & u_3^{(L)} \\ \vdots & \vdots & \ddots & \vdots \\ u_L^{(1)} & u_L^{(2)} & \cdots & u_L^{(L)} \end{bmatrix} \begin{bmatrix} \lambda_{11} & 0 & \cdots & 0 \\ 0 & \lambda_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_{LL} \end{bmatrix} \begin{bmatrix} u_1^{(1)} & u_1^{(2)} & \cdots & u_1^{(L)} \\ u_2^{(1)} & u_2^{(2)} & \cdots & u_2^{(L)} \\ u_3^{(1)} & u_3^{(2)} & \cdots & u_3^{(L)} \\ \vdots & \vdots & \ddots & \vdots \\ u_L^{(1)} & u_L^{(2)} & \cdots & u_L^{(L)} \end{bmatrix}^\top\end{aligned} \tag{3.6}$$

From this representation, the original trajectory matrix can be decomposed into $q = \max\{i|\lambda_{ii} > 0\} \leqslant L$ matrices, by the projection of each column in $\mathbf{X_m}$ onto the respective eigenvector in $\mathbf{U}$:

$$\begin{aligned}\mathbf{X_m} &= \sum_{i=1}^{q\leqslant L} \mathbf{u}_i(\mathbf{x}_i^\top \mathbf{u}_i)^\top = \sum_{i=1}^{q\leqslant L} \mathbf{u}_i \mathbf{x}_{pc_i}^\top \\ &= \underbrace{\begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \\ \vdots \\ u_L^{(1)} \end{bmatrix} \begin{bmatrix} x_{pc_1}^{(1)} \\ x_{pc_2}^{(1)} \\ \vdots \\ x_{pc_K}^{(1)} \end{bmatrix}^\top}_{\mathbf{X_{m_1}}} + \cdots + \underbrace{\begin{bmatrix} u_1^{(q)} \\ u_2^{(q)} \\ \vdots \\ u_L^{(q)} \end{bmatrix} \begin{bmatrix} x_{pc_1}^{(q)} \\ x_{pc_2}^{(q)} \\ \vdots \\ x_{pc_K}^{(q)} \end{bmatrix}^\top}_{\mathbf{X_{m_q}}}\end{aligned} \tag{3.7}$$

where $\mathbf{x}_{pc_i} \in \mathbb{R}^{K\times L}$ is the i-th principal component of the time series.

Resulting in the same set of decomposed matrices $X_\mathcal{D}$ as in Equation 3.4.

2. **Reconstruction**

a) **Grouping**:

In this step, the matrices of $X_\mathcal{D}$ are merged into disjoint groups. Defining $\mathcal{I} = \{I_1, I_2, \ldots, I_i, \ldots, I_p\}$ such that a given set of indices $I_i = \{i_1, i_2, \ldots, i_{p_i}\}$ corresponds to the matrix indices that belongs to the i-th group. Then, the grouping step is perform as in Equation 3.8, by adding up all the matrices that belongs to that group.

$$\mathbf{X}_{I_i} = \sum_{i\in I_i} \mathbf{X}_{mi} = \mathbf{X}_{mi_1} + \mathbf{X}_{mi_2} + \cdots + \mathbf{X}_{mi_{p_i}} \tag{3.8}$$

Extending to all sets of indices results in a set of grouped matrices:

$$X_\mathcal{I} = \{\mathbf{X}_{I_1}, \mathbf{X}_{I_2}, \ldots, \mathbf{X}_{I_p}\} \tag{3.9}$$

This step relies on prior knowledge about the separability of the components, the magnitude of the eigenvalues, the existence of patterns between the eigenvectors, among others. Roughly speaking, at this step the filtering effectively occurs, by dismissing some matrices. The reader is referred to (GOLYANDINA *et al.*, 2013, p. 43) and (GOLYANDINA; KOROBEYNIKOV, 2014, p. 12-15) for a detailed discussion about these aspects.

b) **Diagonal Averaging**:

This step aims to map each matrix from the grouping step in $X_{\mathcal{I}}$ back into a one-dimensional time series, $\tilde{X}_N$, with the same length as the original:

$$\tilde{X}_N^{(k)} = \begin{pmatrix} \tilde{x}_0^{(k)} & \tilde{x}_1^{(k)} & \dots & \tilde{x}_n^{(k)} & \dots & \tilde{x}_{N-2}^{(k)} & \tilde{x}_{N-1}^{(k)} \end{pmatrix} \tag{3.10}$$

Let $\mathbf{X_{I_i}} = \mathbf{A} = (a_{ij})_{i,j=1}^{L,K}$ represent a matrix from the grouping step. The k-th reconstructed time series in Equation 3.10 is obtained by performing an average operation on each anti-diagonal of $\mathbf{A}$. This operation is shown in Equation 3.11 and Figure 7, and it is repeated for $n = 0, 1, \dots, N-1$.

$$\tilde{x}_n^{(k)} = \begin{cases} \dfrac{1}{n+1} \displaystyle\sum_{i=1}^{n+1} a^*_{ij'}, & 0 \leqslant n < L^* - 1 \\ \dfrac{1}{L^*} \displaystyle\sum_{i=1}^{L^*} a^*_{ij'}, & L^* - 1 \leqslant n < K^* \\ \dfrac{1}{N-n} \displaystyle\sum_{i=n-K^*+2}^{N-K^*+1} a^*_{ij'}, & K^* \leqslant n < N \end{cases} \tag{3.11}$$

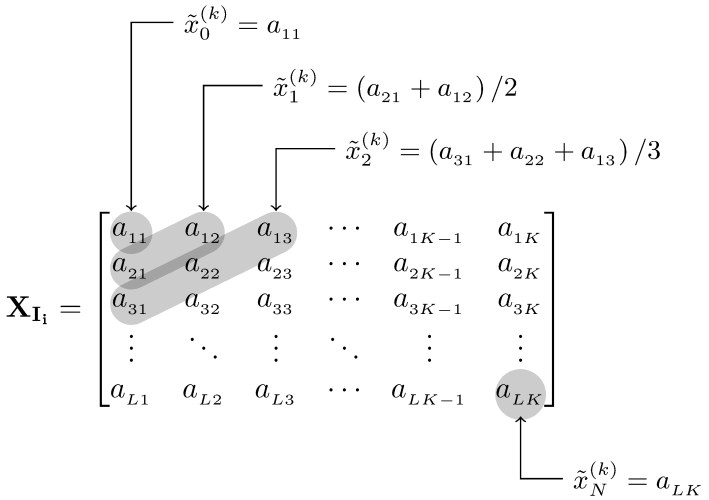


Figure 7 – Diagonal averaging process.

where $j' = n - i + 2$, $L^* = \min(L, K)$, $K^* = \max(L, K)$, $N = L + K - 1$, $a^*_{ij} = a_{ij}$ for $L < K$, and $a^*_{ij} = a_{ji}$ for $L \geqslant K$.

Extending this procedure for each component $k = 1, 2, \ldots, q \leqslant L$, the original time series is approximated by its reconstructed version $\tilde{X}_N^R$:

$$\tilde{X}_N^R = \sum_{k=1}^{R} \tilde{X}_N^{(k)} \tag{3.12}$$

where R is a parameter that controls the number of components used to approximate the time series, with $1 \leqslant R \leqslant L$.

If $R \equiv L$, then $\tilde{X}_N^R \equiv X_N$, the reconstructed series is equal to the original one, if $R < L$, then $\tilde{X}_N^R$ is an approximation of X_N considering the first R components. The SSA filtering steps are summarized in the pseudo-algorithm shown in Algorithm 1. Note that (3.12) is just one specific example of the possible reconstructions permitted by the SSA method. Any combination of components can be chosen for reconstruction, not just the first principal components or sequential components.

Algorithm 1 – SSA Algorithm

Input: $X_N \in \mathbb{R}^N$: original time series
$2 \leqslant L \leqslant N/2 \in \mathbb{Z}$: embedding dimension
$1 \leqslant R \leqslant L \in \mathbb{Z}$: number of components for reconstruction
Output: $\tilde{X}_N^R \in \mathbb{R}^N$: reconstructed time series

1: **procedure** SSA(X_N, L, R)
2: $\mathbf{X_m} \leftarrow$ Embedding(X_N, L) ▷ Figure 6
3: $X_{\mathcal{D}} \leftarrow$ Decomposition$(\mathbf{X_m})$ ▷ Equation 3.3 or 3.7
4: $X_{\mathcal{I}} \leftarrow$ Grouping$(X_{\mathcal{D}}, \mathcal{I})$ ▷ Equation 3.8 and 3.9
5: $\tilde{X}_N^R \leftarrow$ DiagonalAveraging$(X_{\mathcal{I}}, R)$ ▷ Equation 3.11
6: **end procedure**

3.2 Real-time SSA

The SSA method requires a batch of samples from the time series to be available. In order to use it in a real-time application, a modified version of it, called CSSA and proposed in (LELES; MOZELLI; GUIMARAES, 2017, p. 4) was adopted.

The method collects samples from a continuous time signal $x_n = x(nT_s)$ with T_s the sampling interval, to form a time series X_N of fixed length N. After that, as new samples arrive, the old ones are discarded, as in a sliding window illustrated in Figure 8.

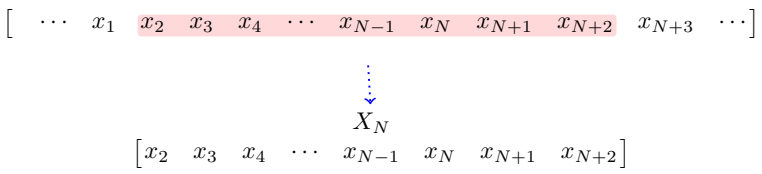


Figure 8 – Sliding window to update the time series.

Besides, for each incoming sample x_n, instead of performing the diagonal averaging on each component's matrix, its reconstruction $\tilde{x}_n$, is done directly from the updated eigenvectors $\mathbf{U}$ after the decomposition step, according to Equation 3.13, which performs the same operation as Equation 3.7, where $u^{(k)}$ are the eigenvectors of the k-th component.

$$\tilde{x}_n^{(k)} = \sum_{i=1}^{L} u_L^{(k)} u_i^{(k)} x_{n-L+i} \qquad (3.13) \qquad\qquad \tilde{x}_n^R = \sum_{k=1}^{R} \tilde{x}_n^{(k)} \qquad (3.14)$$

The reconstructed sample considering then the first R components of the time series is obtained by Equation 3.14. The pseudo-algorithm of this filter is presented in Algorithm 2.

Algorithm 2 – CSSA Algorithm

Input: $x_n \in \mathbb{R}$: sample of the original time series
$N \in \mathbb{Z}$: minimum number of samples to form a time series
$2 \leqslant L \leqslant N/2 \in \mathbb{Z}$: embedding dimension
$1 \leqslant R \leqslant L \in \mathbb{Z}$: number of components for reconstruction
Output: $\tilde{x}_n^R \in \mathbb{R}$: reconstructed sample

procedure CAUSALSSA(x_n, N, L, R)
  $n \leftarrow 0$, $X_N \leftarrow 0$
  while new sample to process **do**
    if $n < N$ **then**
      $X_N \leftarrow$ AddSample(X_N, x_{ns})
      $\tilde{x}_n^R \leftarrow x_n$
    else if $n == N$ **then**
      $\tilde{X}_N^R \leftarrow$ SSA(X_N, L, R) ▷ Algorithm 1
    else if $n > N$ **then**
      $X_N \leftarrow$ UpdateTimeSeries(X_N, x_n) ▷ Figure 8
      $\mathbf{X_m} \leftarrow$ Embedding(X_N, L) ▷ Figure 6
      $\mathbf{C_m} \leftarrow \left(\mathbf{X_m X_m^\top}\right)/K$ ▷ Equation 3.5
      $\mathbf{U} \leftarrow$ Eigendecomposition$(\mathbf{C_m})$ ▷ Equation 3.7
      $\tilde{x}_n^R \leftarrow$ Reconstruction$(X_N, \mathbf{U})$ ▷ Equation 3.13 and 3.14
    end if
    $n \leftarrow n + 1$
  end while
end procedure

In the CSSA algorithm, n and X_N are global variables that maintain its state between function calls and are initialized once. The filter executes as long as new samples arrive. For $n < N$, the filter is inactive. When $n = N$, the minimum number of samples has been reached, then the SSA filter is executed on X_N. After that, for $n > N$, the series is updated with the new sample, the eigendecomposition is performed on the new covariance matrix and the operations in Equation 3.13 and Equation 3.14 are executed to compute the reconstructed sample.

3.3 Example: Applying SSA in a Time Series

In this example, the SSA method is applied to the time series of daily reported cases of Covid-19 in Brazil from Feb. 2020 to Feb. 2022, in Figure 9, for an embedding dimension of $L = 40$. Figure 10 shows the reconstruction of the time series for $R = 1, 2, 5$ and 15 components.

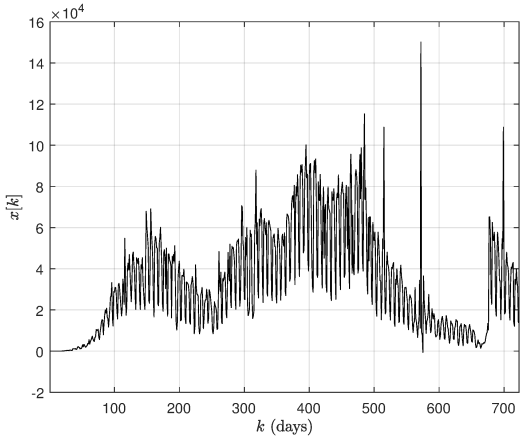


Figure 9 – Time series of Covid-19 daily reported cases in Brazil from Feb./2020 to Feb./2022. Source: Covid-19 (2022).

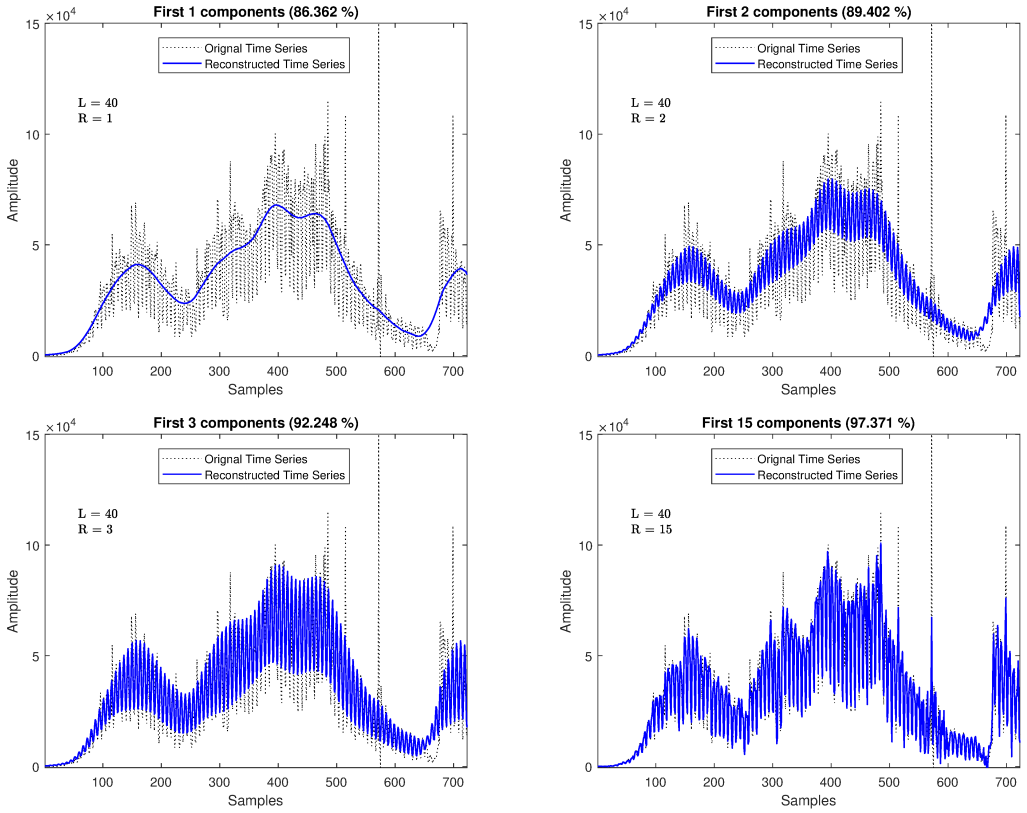


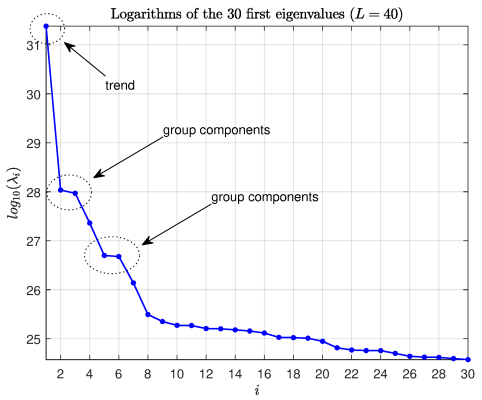
Figure 10 – Time series and its reconstruction with fewer components from the SSA decomposition.

The first component represents the time series' trend, and by adding more compo-

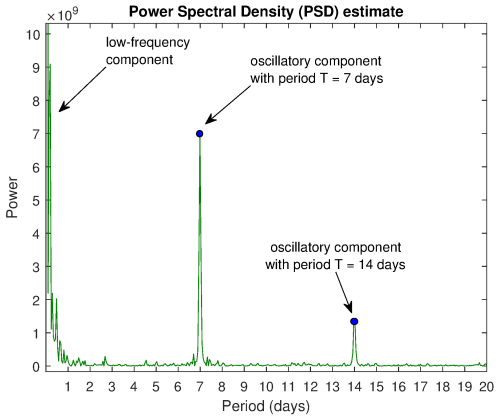
nents, by increasing $R \uparrow$, the reconstructed version gets closer to the original time series. Since the components identified by the SSA are ordered according to the magnitude of their corresponding eigenvalues or singular values, the highest one is associated with the trend, which alone represents $\approx 86\%$ of the original time series in this example, as presented in Figure 10 (a).

Figure 11 (a) shows the magnitude of the component's eigenvalues. The first eigenvalue is the weight of the trend. Similar values in the magnitude plot indicate pair of components that should be grouped, and they usually represent oscillatory components, which is the case for components 2-3 and 5-6. The eigenvalues with small magnitudes are associated with components with less spectral density and are usually of high frequency, such as noise.

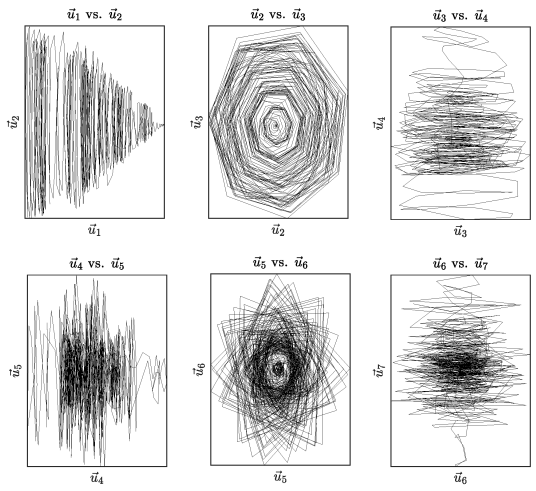
Further visual information usueful for the grouping step and reconstruction can be extracted from the Periodogram, the pairwise scatterplot of the eigenvectors, and the component's correlation matrix, shown in Figure 11 (b)-(d).



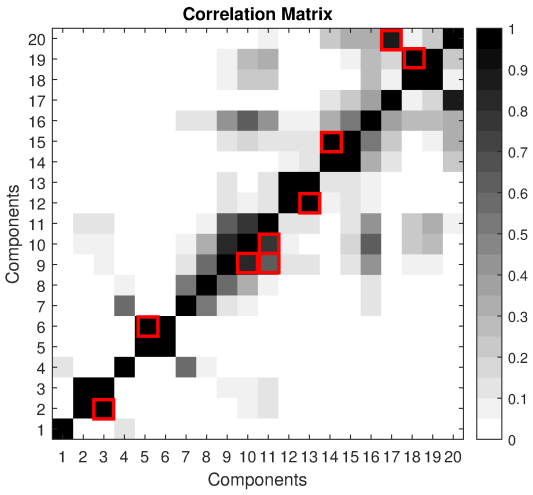
(a) Magnitude of the component's eigenvalues.



(b) Periodogram.



(c) Eigenvector's pairwise scatterplot.



(d) Component's correlation matrix.

Figure 11 – SSA method: visual information used to group and select components for reconstruction

The Periodogram indicates in which periods or frequencies the time series' compo-

nents are located. Figure 11 (b) shows three components: the low-frequency trend, and two oscillatory components with a period of 7 and 14 days. The pairwise scatterplot of the eigenvectors shows which components are related. In Figure 11 (c), the components 2-3 and 5-6 are clearly correlated, and they represent the oscillatory components shown in the Periodogram of the time series. The period of oscillation is the number of edges of the polygon formed by the pairs of eigenvectors. The correlation matrix in Figure 11 (d) show the degree of correlation between the time series components. Useful in the grouping step and in verifying the embedding dimension's choice.

For a detailed explanation of parameter selection for the Singular Spectrum Analysis (SSA), please refer to Golyandina (2010), Golyandina *et al.* (2013) and more recently Leles (2017).

3.4 Filter Bank SSA

The SSA method can also be viewed in terms of a linear filtering process. The projection operation that results in a set of decomposed matrices $X_{\mathcal{D}}$ through Equation 3.3 or Equation 3.7, in the decomposition and the diagonal averaging steps, leads to the description of the SSA as a FB of Figure 12.

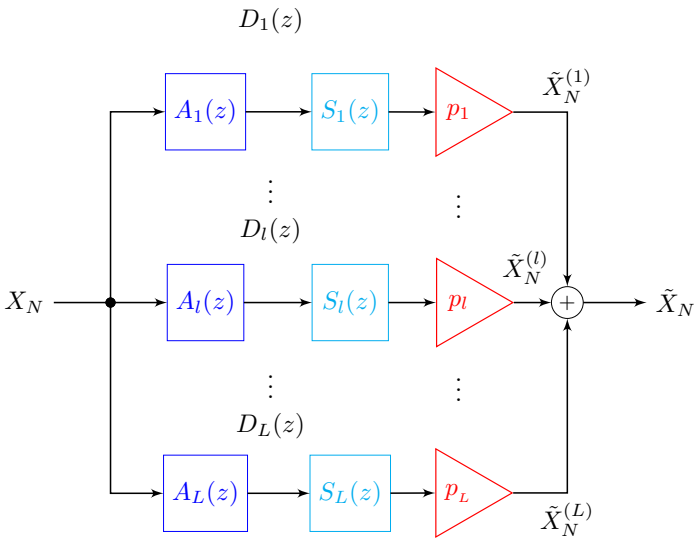


Figure 12 – Block diagram of the SSA as a FIR FB.

The eigenvectors from the trajectory matrix decomposition form the coefficients of the filters $A_l(z)$ and $S_l(z)$, for each component $l = 1, 2, \ldots, L$. The TF of $A_l(z)$ in Equation 3.15 is a causal filter and represents the projection operation that results in the principal components, a process called forward filtering. Meanwhile, the TF of $S_l(z)$ in Equation 3.16, is an anti-causal filter and is responsible for the inverse projection, and also

the diagonal averaging step, by normalizing by L, performing the reverse filtering process, where $\mathbf{u}^{(l)} = \left[u_0^{(l)}\, u_1^{(l)}\, \dots\, u_k^{(l)}\, \dots\, u_{L-1}^{(l)}\right]^\top$ are the eigenvectors of the l-th component.

$$A_l(z) = \sum_{k=0}^{L-1} u_k^{(l)} z^{-k} \tag{3.15}$$

$$S_l(z) = \frac{1}{L} \sum_{k=0}^{L-1} u_k^{(l)} z^{k} \tag{3.16}$$

The binary gain $p_l \in \{0, 1\}$ in Figure 13 indicates whether or not a given component is selected for the reconstruction of the time series. The l-th branch of the FB, for $p_l = 1$, relates the l-th reconstructed component $\tilde{X}_N^{(l)}$ with the original time series X_N through the transfer function of a FIR filter $D_l(z)$ given by Equation 3.17.

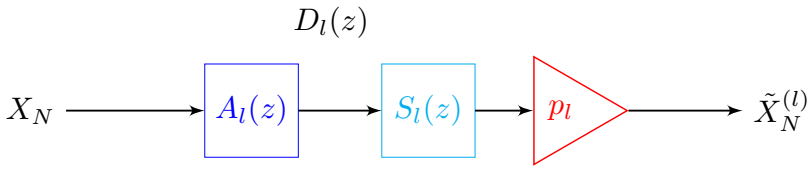


Figure 13 – SSA FB: l-th eigenfilter $D_l(z) = A_l(z)S_l(z)p_l$.

$$\begin{aligned} D_l(z) = \frac{\tilde{X}_N^{(l)}(z)}{X_N(z)} = A_l(z)S_l(z) &= \left(\sum_{k=0}^{L-1} u_k^{(l)} z^{-k}\right)\left(\frac{1}{L}\sum_{k=0}^{L-1} u_k^{(l)} z^{k}\right) \\ &= \sum_{k=-(L-1)}^{L-1} d_k^{(l)} z^k = \sum_{k=1-L}^{L-1}\left(\frac{1}{L}\sum_{i=|k|}^{L-1} u_i^{(l)} u_{|k|-i}^{(l)}\right) z^k \end{aligned} \tag{3.17}$$

According to Tomé *et al.* (2018), the filter coefficients $d_k^{(l)}$ are the entries of the matrix $\mathbf{D}_l = \mathbf{u}^{(l)}\mathbf{u}^{(l)^\top}$ along the k-th diagonal. The main diagonal is given by $k = 0$, and above and below the main diagonal by $k > 0$ and $k < 0$, respectively. $\mathbf{D}_l$ is a symmetric matrix, $d_{-k} = d_k$, and the sum of each matrix component results in an identity matrix as shown in Equation 3.18, where $\delta(k)$ is the unit impulse signal.

$$\sum_{l=1}^{L} \mathbf{D}_l = \sum_{l=1}^{L} \mathbf{u}^{(l)}\mathbf{u}^{(l)^\top} = \mathbf{I}_{L\times L} \implies \sum_{l=1}^{L} d_k^{(l)} = \delta(k) \tag{3.18}$$

This implies that if the reconstruction is done with all the components, $p_l = 1, \forall_{l=1,\dots,L}$, then the original time series is exactly recovered:

$$\begin{aligned} \tilde{X}_N(z) = X_N(z)\sum_{l=1}^{L} D_l(z) &= X_N(z)\left(\sum_{l=1}^{L}\left(\sum_{k=1-L}^{L-1} d_k^{(l)} z^k\right)\right) \\ = X_N(z)\left(\sum_{k=1-L}^{L-1}\left(\sum_{l=1}^{L} d_k^{(l)}\right) z^k\right) &= X_N(z)\left(\sum_{k=1-L}^{L-1} \delta(k) z^k\right) = X_N(z) \end{aligned} \tag{3.19}$$

Which, in the time-domain, is equivalent to convolve the time series with an impulse signal. Additionally, the symmetry property allows a closed-form description of the filter's

frequency response $D_l(z)$, as described in Equation 3.20-3.23.

$$D_l(z) = d_0^{(l)} + \sum_{m=1-L}^{-1} d_m^{(l)} z^m + \sum_{k=1}^{L-1} d_k^{(l)} z^k, \quad (m \leftarrow -k) \tag{3.20}$$

$$= d_0^{(l)} + \sum_{k=1}^{L-1} \left(d_{-k}^{(l)} z^{-k} + d_k^{(l)} z^k \right), \quad \left(d_{-k} \leftarrow d_k, \quad z \leftarrow e^{j\omega} \right) \tag{3.21}$$

$$D_l(e^{j\omega}) = d_0^{(l)} + \sum_{k=1}^{L-1} d_k^{(l)} \left(e^{j\omega k} + e^{-j\omega k} \right), \quad \left(e^{j\omega k} + e^{-j\omega k} \leftarrow 2\cos(k\omega) \right) \tag{3.22}$$

$$D_l(e^{j\omega}) = d_0^{(l)} + 2\sum_{k=1}^{L-1} d_k^{(l)} \cos(k\omega), \quad \forall\, l = 1, 2, \ldots, L \tag{3.23}$$

Representing the frequency response of a real-valued and zero-phase filter, which implies in no signal distortion and that each component is in phase with the original time series.

Therefore, regarding the reconstruction stage, the SSA can be viewed as a linear filtering process performed through a bank of FIR filters generated by the spectral decomposition of the time series in terms of its eigenvectors. For a more detailed characterization of the SSA in the frequency-domain, the reader is referred to Tomé *et al.* (2018), Kume (2012), Harris and Yuan (2010) and Leles *et al.* (2016).

3.5 Example: Applying SSA in the Frequency-Domain

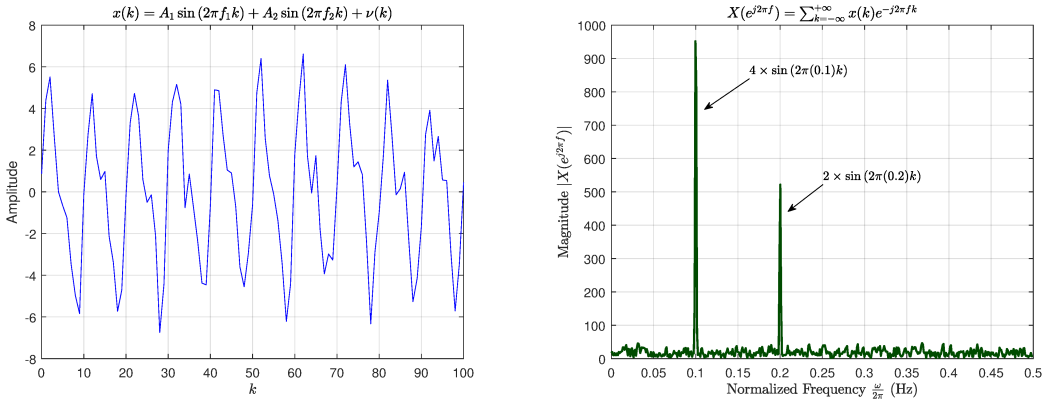
To analyze the SSA method in the frequency-domain, the time series in Equation 3.24 is considered for $N = 1024$ samples. It consists of a sum of two sinusoids with amplitudes 4 and 2 and frequencies 0.1 Hz and 0.2 Hz, mixed with gaussian noise with zero mean $\mu = 0$ and unit standard deviation, $\sigma^2 = 1$. Figure 14 shows a snippet of the time series and its Discrete-Time Fourier Transform (DTFT).

$$X_N = 4 \times \sin(2\pi(0.1)k) + 2 \times \sin(2\pi(0.2)k) + \nu(k) \tag{3.24}$$

Figure 15 shows the the eigenfilters $D_l(e^{j2\pi f}) = A_l(e^{j2\pi f})S_l(e^{j2\pi f})$, $l = 1, 2, \ldots, L$, where $\omega = 2\pi f$, for decomposing the time series into $L = 5, 10, 20$ and 30 components. By increasing the parameter L, the frequency response of the filters becomes narrower, thus, more precise for a given frequency where a component of the time series might be located.

The impulse response $d_k^{(l)} \equiv d^{(l)}(k)$ of the eigenfilters $l = 1, 5, 9$ and 20 are presented in Figure 16. They represent filters with Finite Impulse Response (FIR) and zero-phase due to their even symmetry, $d^{(l)}(-k) \equiv d^{(l)}(k)$.

The linear filtering process, as described in the FB description of the SSA in Figure 12, is illustrated in Figure 17 to extract the oscillatory and noisy components of the time



(a) Original time series. (b) DTFT of the time series.

Figure 14 – Time series in the time and frequency domain.

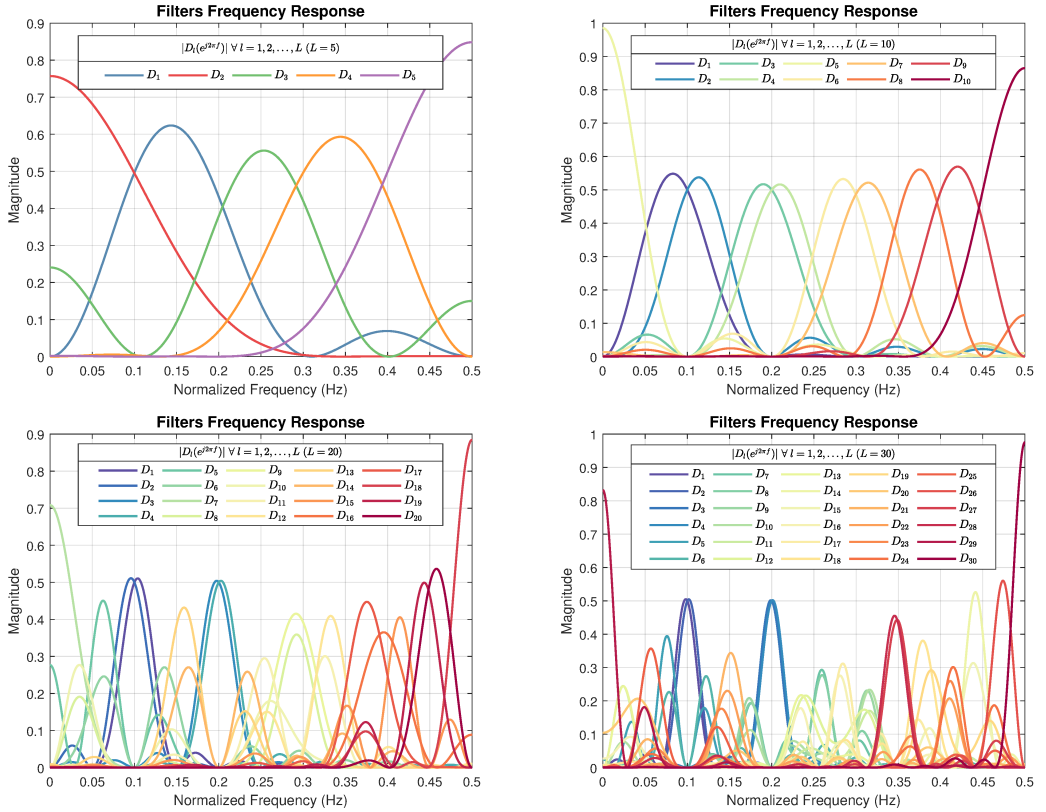


Figure 15 – SSA FB: Frequency response of the time series eigenfilters $D_l(e^{j2\pi f})$, $l = 1, 2, \ldots, L$, for $L = 5, 10, 20$ and 30.

series. In each sub-figure in Figure 17, the upper graph shows the magnitude of the DTFT $|X(e^{j\omega})|$ of the time series X_N, overlapped by the magnitude of the eigenfilters frequency response $|D_l(e^{j\omega})|$. While the lower graph shows the filtered or reconstructed component $\tilde{X}_N^{(l)}$, compared with the original time series X_N in the time-domain.

In Figure 17 (a), by grouping the eigenfilters $|D_1(e^{j\omega})|$ and $|D_2(e^{j\omega})|$, the first component $\tilde{X}_N^{(1)}$, which corresponds to the sinusoid of amplitude 4 and frequency 0.1 Hz, is extracted from the original time series. Similarly, the second oscillatory component is re-

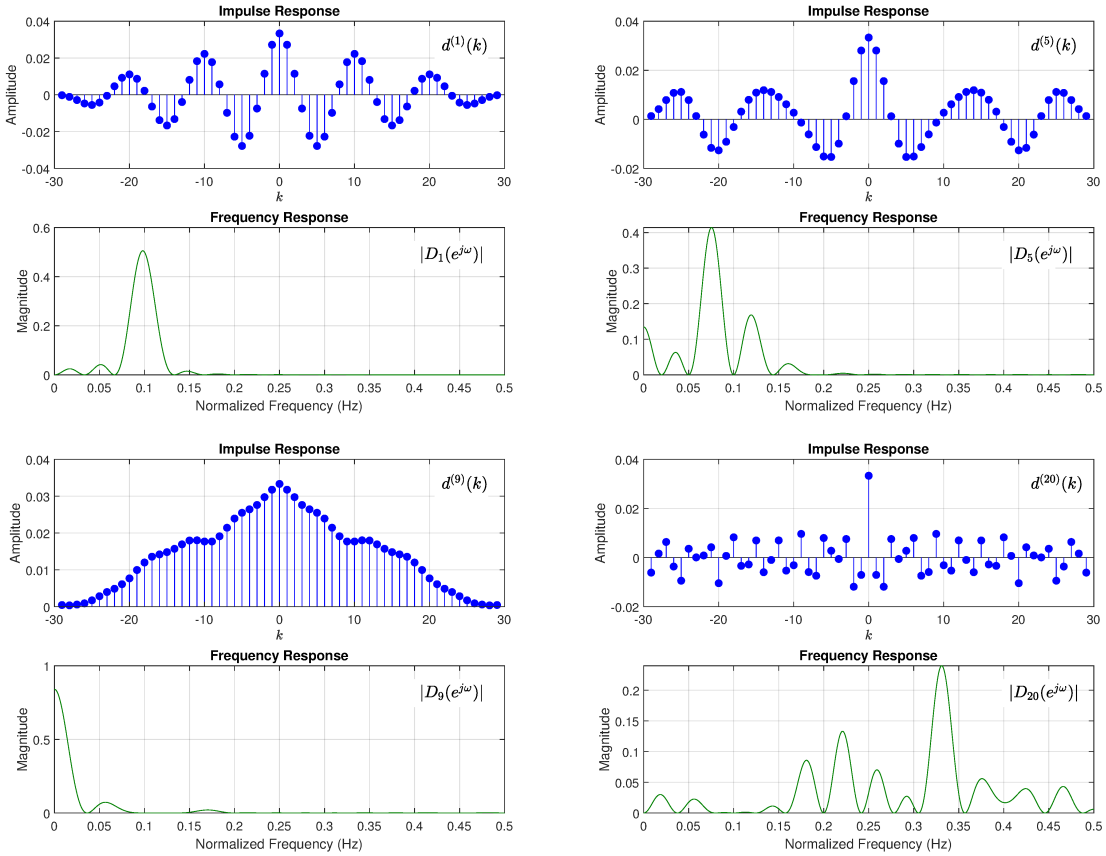
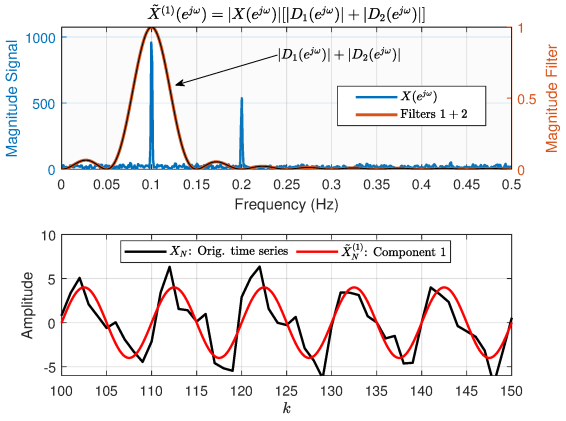
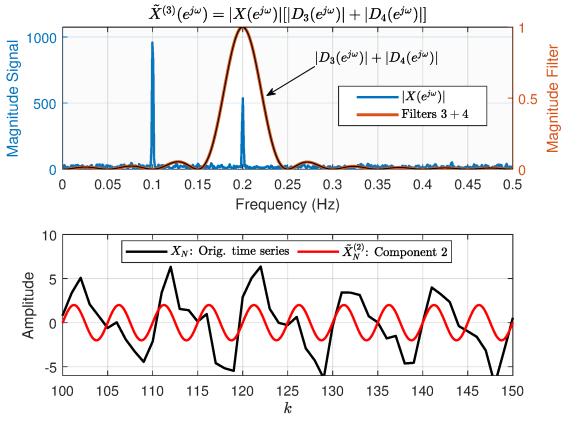


Figure 16 – SSA FB: Impulse and freq. response of $D_l(e^{j2\pi f})$ for $l = 1, 5, 9, 20$ $(L = 30)$.

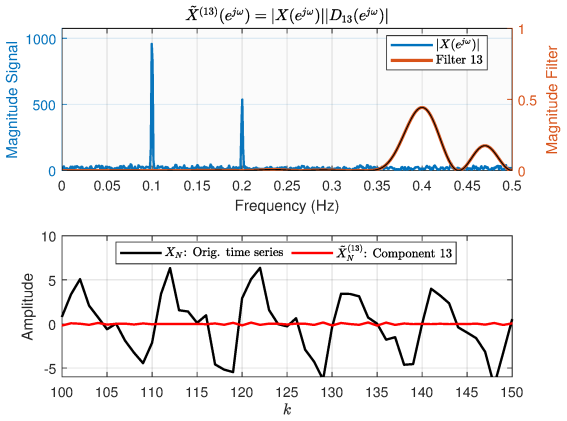
trieved by summing the third and forth eigenfilters $|D_3(e^{j\omega})| + |D_4(e^{j\omega})|$ and multiplying it by $|X(e^{j\omega})|$ in Figure 17 (b). The noise added to the time series is extracted by the remaining filters, as in Figure 17 (c) for the $|D_{13}(e^{j\omega})|$, and in Figure 17 (d) for $\sum_{l=11}^{20} |D_l(e^{j\omega})|$.



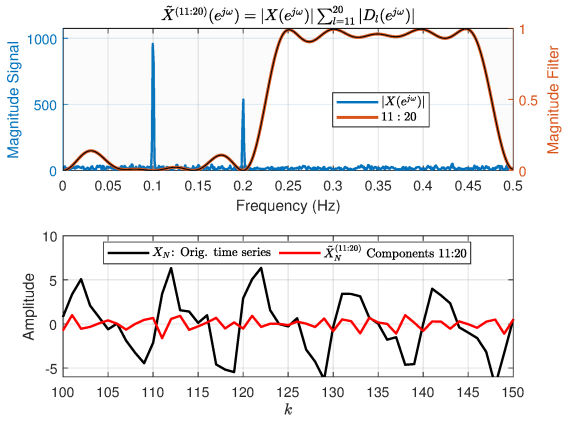
(a) Recovering $4 \times \sin(2\pi(0.1)k)$.



(b) Recovering $2 \times \sin(2\pi(0.2)k)$.



(c) Noisy component.



(d) Low/high-frequency noisy components.

Figure 17 – SSA FB: Filtering the oscillatory and noisy components from the time series ($L = 20$).

CHAPTER

4

RESULTS

This chapter presents the results regarding the use of the SSA method to attenuate measurement noise in PID control loops and the use of the SSA in a trajectory filter generating approach. The SSA for PID control describes the processes modeling, the controller synthesis, the performance measurement metrics, and the numerical simulations analyzing the performance and the measurement noise attenuation for different scenarios and process dynamics. Finally, the numerical simulations of the SSA trajectory generating filter are presented.

4.1 SSA in PID Control

The block diagram in Figure 18 was considered to analyze the effects of applying the SSA noise filter in the performance and sensitivity to measurement noise of the closed-loop system with a PID controller.

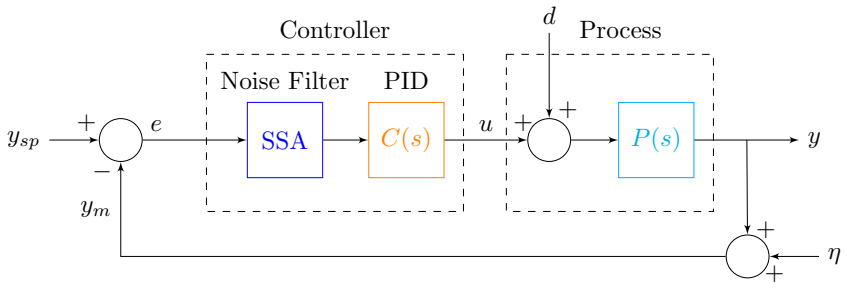


Figure 18 – Block diagram of the closed-loop system analyzed: SSA noise filter, process $P(s)$, ideal PID controller $C(s)$, set-point y_{sp}, error e, control signal u, load disturbance d, process variable y, and measurement noise η.

It consists of a non-parametric SSA based noise filter, in series with an ideal PID controller, that manipulates the process $P(s)$ while subjected to a load disturbance d, at the

process input, by comparing the measured value of the output y_m, mixed with noise η, to the set-point y_{sp}.

4.1.1 Process Dynamics and Modeling

The processes $P(s)$ used in Figure 18 were based on Segovia, Hägglund and Åström (2013), Segovia, Hägglund and Åström (2014) and cover a significant range of dynamics usually encountered in process control. They are: (i) lag-dominated P_L in Equation 4.1; (ii) balanced P_B in Equation 4.2, and (iii) delay-dominated dynamics P_D in Equation 4.3.

$$P_L(s) = \frac{1}{(s+1)(0.1s+1)(0.01s+1)(0.001s+1)} \tag{4.1}$$

$$P_B(s) = \frac{1}{(s+1)^4} \tag{4.2}$$

$$P_D(s) = \frac{e^{-s}}{\left(\frac{5}{100}s+1\right)^2} \tag{4.3}$$

For the controller design, the processes were approximated by the FOTD model in Equation A.1.

$$G(s) = \frac{k_p e^{-\tau_d s}}{\tau s + 1} \tag{A.1}$$

The response curve of each process to a unit step input defined the parameters k_p, τ_d, and τ. The gains of the step response and the unit step determined the gain k_p. The time delay τ_d was obtained through the intersection of the tangent line at the inflection point of the response curve with the time axis, a procedure similar to the one shown in Figure 37 (b). The time constant was given by $\tau = t_{63\%} - \tau_d$, where $t_{63\%}$ is the instant when the response reaches $\approx 63\%$ of its steady-state value. Their values are presented in Table 1.

Table 1 – FOTD parameters' model for lag, balanced and delay-dominated process dynamics.

	$P_L(s)$	$P_B(s)$	$P_D(s)$
k_p	1	1	1
τ_d	0.08	1.43	1.01
τ	1.04	2.92	0.09

4.1.2 Performance Measurement

As in Segovia, Hägglund and Åström (2014), the Integral Absolute Error (IAE) metric in Equation 4.4 was chosen to measure the performance of the controller in the closed-loop system of Figure 18.

$$IAE = \int_0^\infty |e(t)|dt \tag{4.4}$$

where e is the error due to a unit step input.

To measure the control activity u and system output y variability due to measurement noise, the Total Variance (TV) metric was chosen according to Huba (2015) and are reproduced in Equation 4.5 and 4.6.

$$TV(u) = \sum_{k} |u(k+1) - u(k)| - |2\bar{u}_m - u_\infty - u_0| \tag{4.5}$$

$$TV(y) = \sum_{k} |y(k+1) - y(k)| - |2\bar{y}_m - y_\infty - y_0| \tag{4.6}$$

where k is the sample index, $\bar{y}_m \notin (y_0, y_\infty)$ and $\bar{u}_m \notin (u_0, u_\infty)$ are extreme values of y and u and chosen as $\bar{y}_m = 10 \times y_m$ and $\bar{u}_m = 10 \times u_m$. Meanwhile, y_0, u_0, y_∞ and u_∞ are the initial and steady-state values of y and u, respectively.

4.1.3 PID Controller Synthesis

The PI and PID controller parameters used in the simulations are shown in Table 2. Which were obtained applying the AMIGO tuning rules in Table 10 on the FOTD model parameters of the processes, presented in Table 1, for a sensitivity of $M_s = 1.4$.

Table 2 – PI and PID controller parameters for lag, balanced and delay-dominated process dynamics with AMIGO tuning rules.

	$P_L(s)$		$P_B(s)$		$P_D(s)$	
	PI	PID	PI	PID	PI	PID
k_c	4.13	6.44	0.41	1.1	0.18	0.24
k_i	7.67	17.86	0.16	0.47	0.47	0.51
k_d	0	0.24	0	0.69	0	0.03

They were projected in the continuous-time given that the sampling period T_s of the simulations are small enough, such that they are valid in the discrete-time tests.

4.1.4 Experimental Results

The simulations of the system in Figure 18 aimed to investigate the behavior of the closed-loop system performance, measured by the IAE, and the control activity due to measurement noise, measured by the TV, as the values of L and R were changed, considering both the presence and absence of measurement noise. In the first scenario, the error signal $e(t)$ was decomposed into a fixed number of components, constant value of L, the embedding dimension parameter, while its reconstruction was made with a different number of components, varying R for each L. In the second scenario, the opposite was analyzed, the number of components R used to reconstruct the signal was kept constant while the value of L was changed.

The simulations lasted $15\,\text{s}$, with sampling period of $T_s = 0.01\,\text{s}$, for a unit step reference input y_{sp} applied at $t = 5\,\text{s}$, with no load disturbance, $d = 0$. The measurement noise η added at the process output y was from a Gaussian distribution with zero mean and Signal-to-Noise Ratio (SNR) of 0.01.

For the CSSA, the non-parametric noise filter, the sliding window's length for the time series was chosen as $N = 200$ samples, as illustrated in Figure 8. The values of L and R for each scenario were: (i) $L = [15, 25, 35, \ldots, 85, 95]$, and R was varied from 1 to 15 components, for each fixed value of L; and (ii) $R = [1, 2, \ldots, 10]$, and L was varied from 10 to 100, for each fixed value of R.

These values of L and R satisfied the filter constraints, $2 \leqslant L \leqslant N/2$ and $1 \leqslant R \leqslant L$, and cover a wide range of possible values of L, given the choice of N. The combination of L and R results in different noise attenuation degrees. Because the reconstruction with a few significant components excludes the high-frequency ones, which are less significant in terms of spectral density, and is usually where the noisy components are located.

4.1.4.1 ***Lag-dominated dynamics***

The results for P_L, lag-dominated dynamics, are shown in Figure 19, 20, 21 and 22. Each point in Figure 19 and 21 represents the value of the IAE and TV of the control action u and system output y, from the simulation of the closed-loop system for a given combination of L and R. Where R_i, $i = 1, 2, \ldots, 15$, means that the error signal e was reconstructed with the first i components, and L_j, $j = 10, 20, \ldots, 100$, means that the error signal e was decomposed into j components.

The results in Figure 19 (a)-(b), in the absence of measurement noise, as the attenuation degree increases, by increasing L and reducing R, both the performance and the signal variability deteriorates, i.e. larger IAE and TV of the control signal u and process variable y, for both PI and PID controllers. The best result is achieved when the filter is not activated, which happens for $L = R = 15$. This suggests that, in the absence of measurement noise, the filter is discarding important information present in the signal when disregarding some of its components.

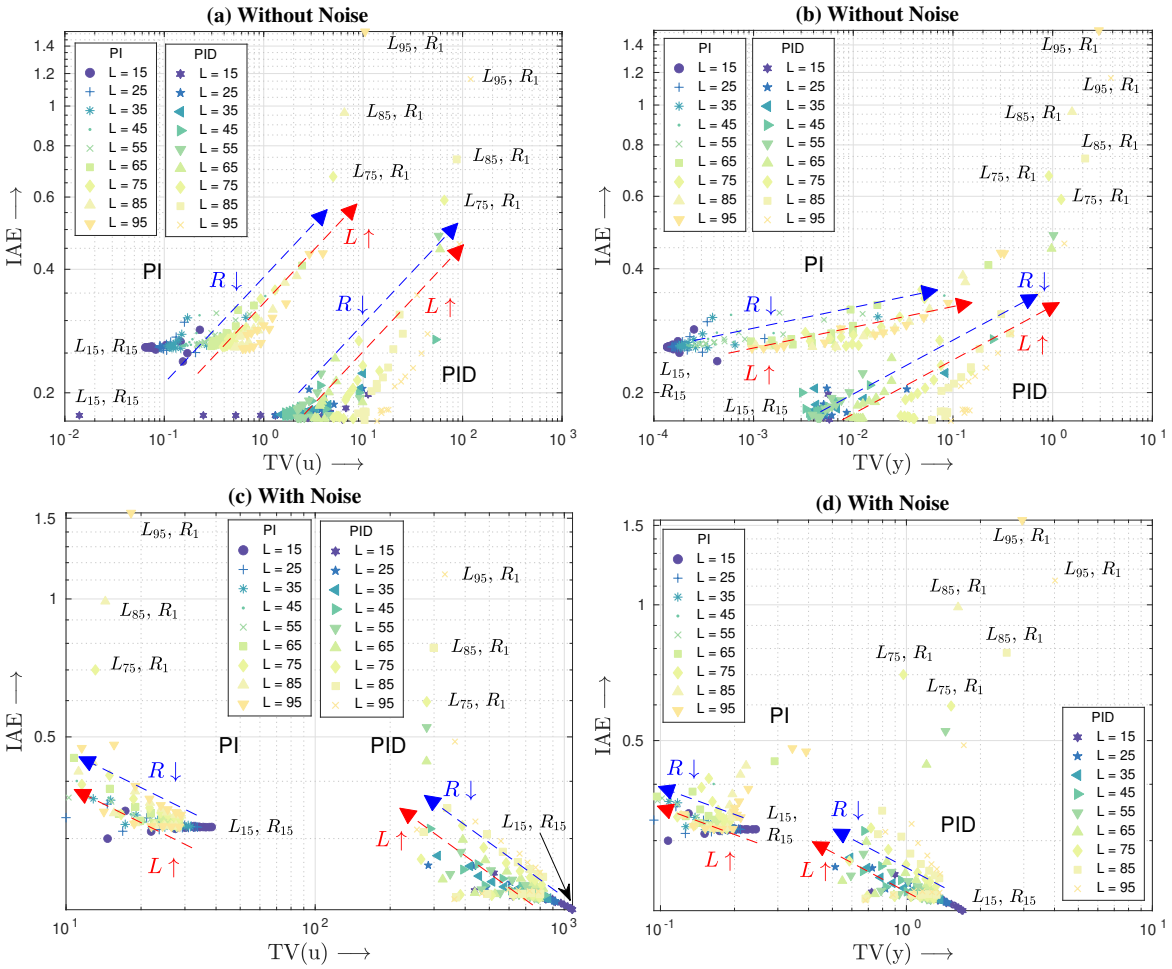


Figure 19 – Performance (IAE) vs. Measurement noise (TV) of the control signal u and process variable y for a lag-dominated dynamics, for a unit step input, SNR = 0.01, $N = 200$, R varying from 1 to 15 for each L, and $T_s = 0.01$ s ($\mathbf{L}$ fixed and $\mathbf{R}$ varying).

In contrast, in the presence of measurement noise, Figure 19 (c)-(d), the higher the attenuation degree, the lower the TV of the control signal u and process variable y, as expected, since the suppression of the noise implies less signal variability, but at the expense of a worse performance, higher IAE, for both controllers. At some point, for the $TV(y)$, greater noise attenuation increases the TV as well. Additionally, the PID controller performed better than the PI for most simulations, but with higher values of TV.

Some of these results can be viewed in the time-domain behavior of the control action u and system output y, for the case where noise is added to the system's output and for some values of L and R, as shown in Figure 20.

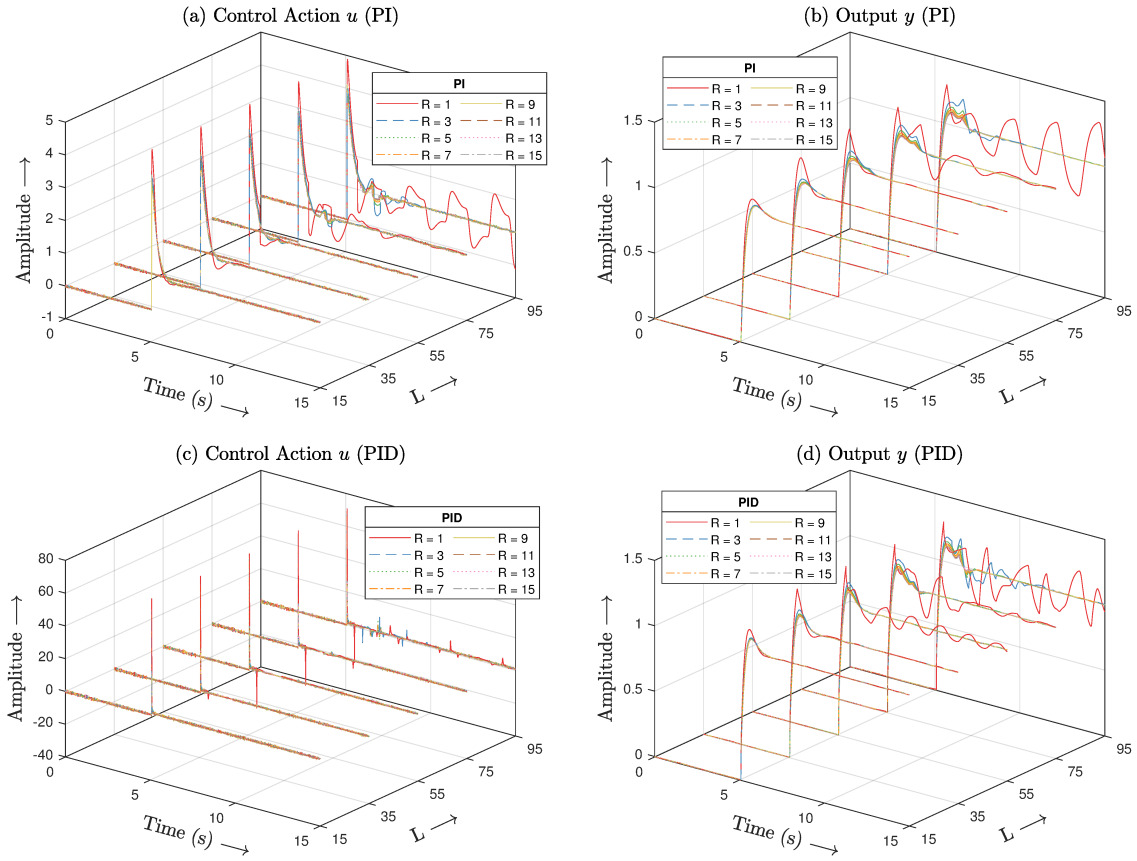


Figure 20 – Closed-loop performance of the lag-dominated dynamics, for a unit step input, in the presence of measurement noise with SNR = 0.01, $N = 200$, R varying from 1 to 15 for each L, and $T_s = 0.01$ s ($\mathbf{L}$ fixed and $\mathbf{R}$ varying).

The worsening of the system's performance can be seen clearly for $R \equiv 1$ as L increases from 15 to 95, which is equivalent to an increase in the degree of noise attenuation. It deteriorates to the point that the system is no longer stable, for the PI controller in Figure 20 (a)-(b), and marginally stable for the PID, see $L \equiv 95$ and $R \equiv 1$ in Figure 20 (c)-(d). Additionally, in Figure 20 the greater variability of the control action for the PID compared to the PI controller, which reflects on the TV metric and is caused by the amplifying effect of the derivative part of the controller in the presence of high-frequency components.

The simulation results for the second scenario, for L varying for each fixed value of R, were similar, as presented in Figure 21 and 22. In Figure 21 (a)-(b), in the absence of measurement noise, less attenuation, higher $R \uparrow$ and lower $L \downarrow$, improves performance and reduces the sensitivity to measurement noise. In contrast, In Figure 21 (c)-(d), in the presence of measurement noise, the trade-off between performance and sensitivity to measurement noise becomes evident. In this case, higher noise attenuation, smaller $R \downarrow$ and higher $L \uparrow$, reduces the TV(u) but increases the IAE.

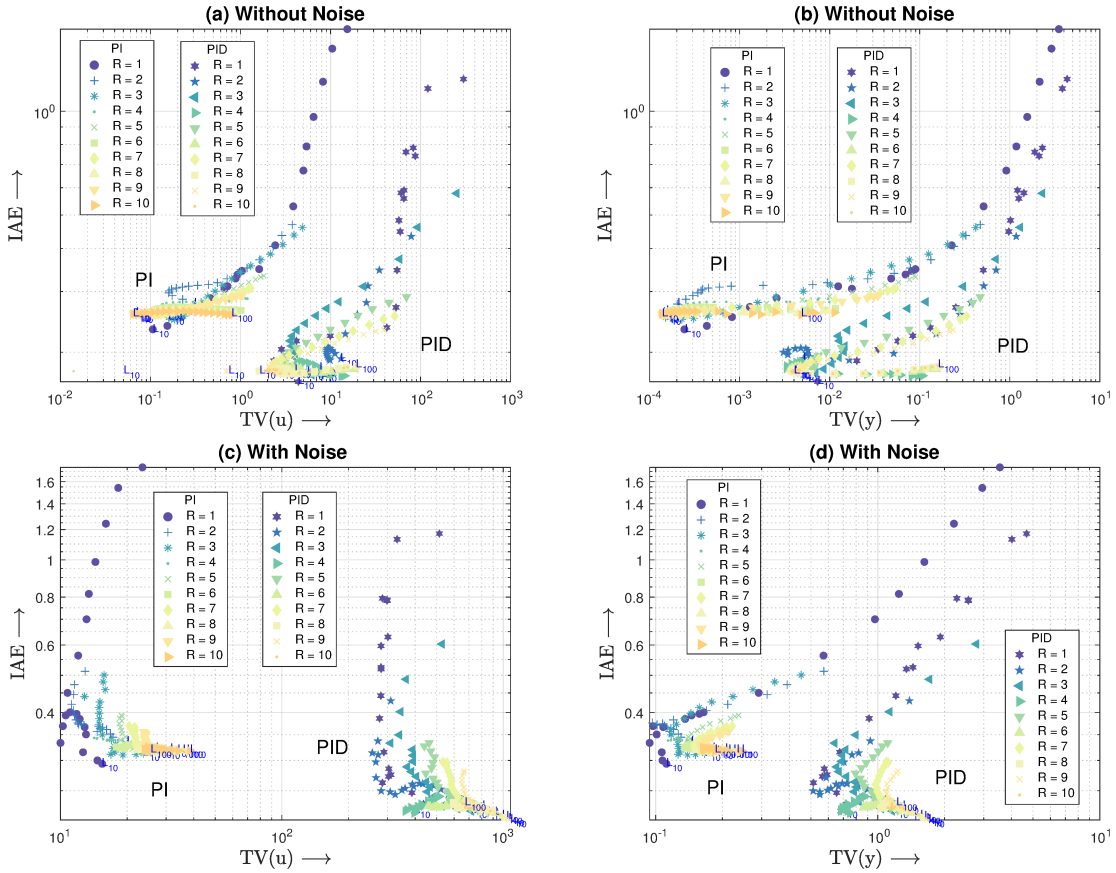


Figure 21 – Performance (IAE) vs. Measurement noise (TV) of the control signal u and process variable y for a lag-dominated dynamics, unit step input, SNR = 0.01, $N = 200$, L varying from 10 to 100 for each R, and $T_s = 0.01$ s ($\mathbf{R}$ fixed and $\mathbf{L}$ varying).

The time behavior of the closed-loop system in Figure 22, varying L for fixed values of R, shows a similar outcome when compared to Figure 20. Again, the PI controller becomes unstable for higher $L \uparrow$ and lower $R \downarrow$, specifically for the highest possible attenuation degree, $L \equiv 100$ and $R \equiv 1$ in Figure 22 (a)-(b). Meanwhile, the PID controller becomes marginally stable for this case, as shown in Figure 22 (c)-(d). Besides, the PID controller showed greater signal variability due to the derivative part.

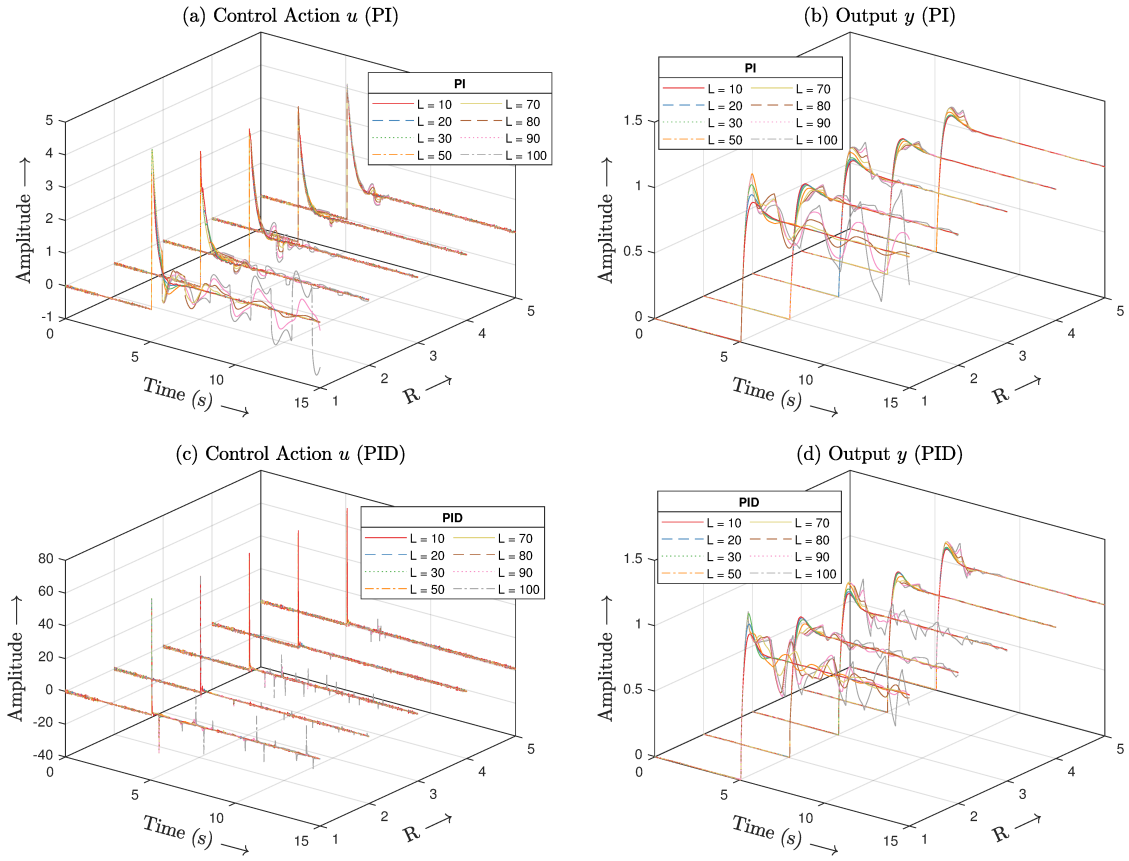


Figure 22 – Closed-loop performance of the lag-dominated dynamics, for a unit step input, in the presence of measurement noise with SNR = 0.01, $N = 200$, L varying from 10 to 100 for each R, and $T_s = 0.01$ s ($\mathbf{R}$ fixed and $\mathbf{L}$ varying).

4.1.4.2 ***Balanced dynamics***

The simulation results for process P_B, of balanced dynamics, are presented in Figure 23, 24, 25 and 26. For better viewing, some results are enlarged. For this process, in the absence of measurement noise and for any value of L and R, the TV of the output y were negligible, as can be seen from the magnitude values on the x-axis of Figure 23 (b). In contrast, they were significant for the control action u in Figure 23 (a) and showed a larger gap between the PI and PID controllers results then the one observed for the lag-dominated dynamics P_L. Moreover, in the absence of measurement noise, the filter deteriorates both the performance and the TV, even for small values of L, as can be seen from the points labeled with R_1. As the degree of noise attenuation is decreased, by making R closer to L, the performance converges to ≈ 4.8 (PI) and ≈ 2.25 (PID), while the TV continues to decrease.

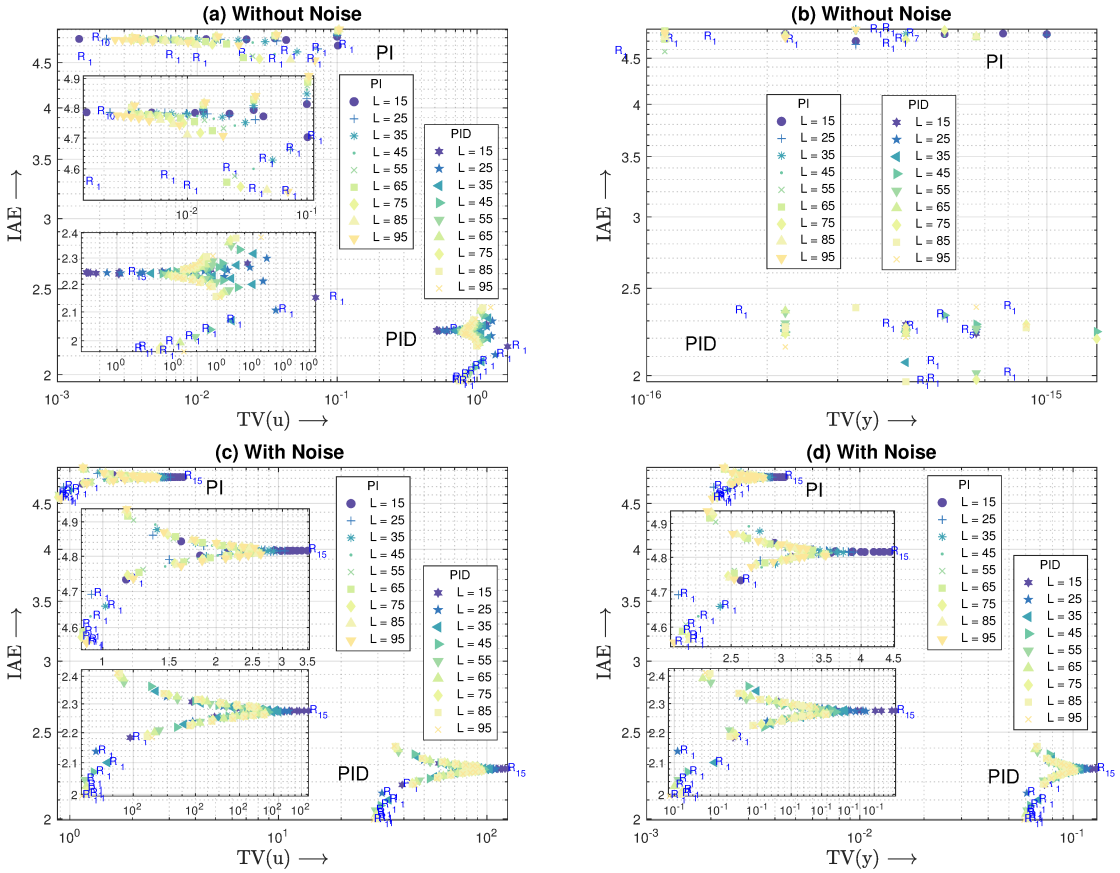


Figure 23 – Performance (IAE) vs. Measurement noise (TV) of the control signal u and process variable y for a balanced dynamics, unit step input, SNR = 0.01, $N = 200$, R varying from 1 to 15 for each L, and $T_s = 0.01$ s ($\mathbf{L}$ fixed and $\mathbf{R}$ varying).

In the presence of measurement noise, Figure 23 (c)-(d), the pattern changes. Although the gap between PI and PID remains large, now, as the attenuation degree increases, moving from right to left, the results split into two curves, one directed upwards and the other downwards, for both the control action u and output y. Showing that, as the attenuation degree decreases, $L \downarrow$ and $R \uparrow$, the values of IAE converges in an oscillatory way for both PI and PID controllers. Suggesting that for this type of dynamics, some values of L and R can achieve better performance and greater noise attenuation, at the same time, even in the presence of measurement noise.

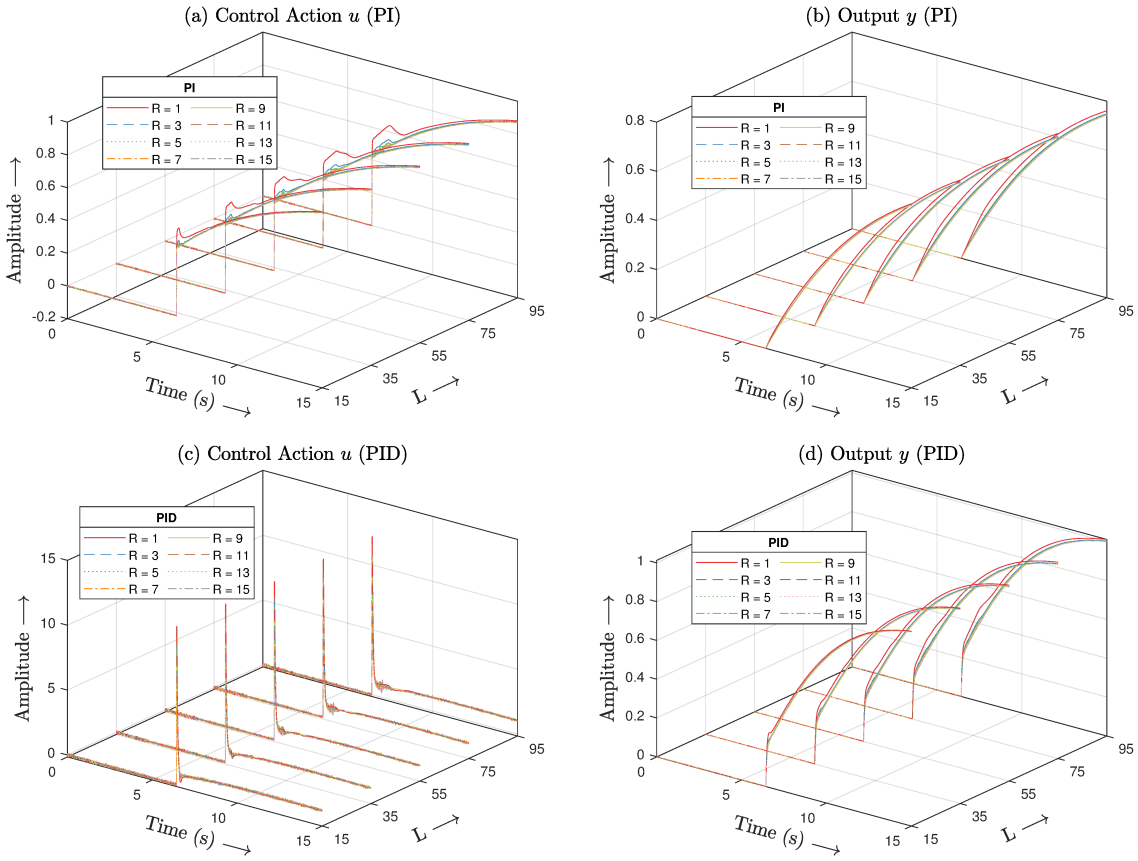


Figure 24 – Closed-loop performance of the balanced dynamics, for a unit step input, in the presence of measurement noise with SNR = 0.01, $N = 200$, R varying from 1 to 15 for each L, and $T_s = 0.01$ s ($\mathbf{L}$ fixed and $\mathbf{R}$ varying).

From the time behavior of the system output y in Figure 24, the PI response is slower than that of the PID, which explains its smaller IAE. Besides, as L and R vary, for this type of process, there are no major differences between the response curves, except for $R \equiv 1$ for the control action of the PI controller in Figure 24 (a). In addition, the magnitude of the control action for the PID was smaller than that of the process P_L of lag-dominated dynamics. It is also worth mentioning that no combination of L and R values resulted in an unstable closed-loop system for the process with balanced dynamics.

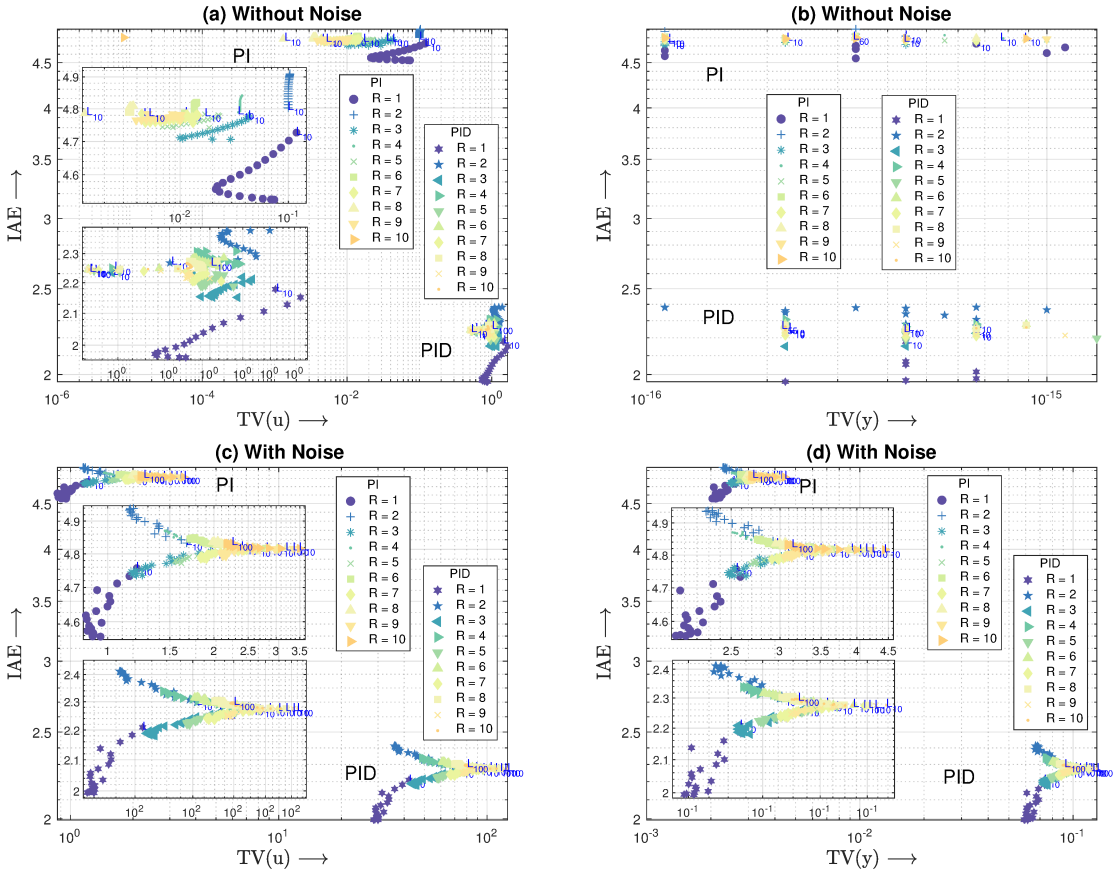


Figure 25 – Performance (IAE) vs. Measurement noise (TV) of the control signal u and process variable y for a balanced dynamics, unit step input, SNR = 0.01, $N = 200$, L varying from 10 to 100 for each R, and $T_s = 0.01$ s ($\mathbf{R}$ fixed and $\mathbf{L}$ varying).

For R fixed and L varying, Figure 25 and 26 show a quite similar behavior. In the absence of measurement noise, the TV(u) is also negligible in Figure 25 (b) as in Figure 23 (b). Differently, the control signal u in Figure 25 (a) does not show any clear pattern, as in Figure 23 (a).

As for the presence of measurement noise, Figure 25 (c)-(d), show more clearly the same oscillatory behavior observed for the IAE in Figure 23 (c)-(d), as the noise attenuation changes. With a large gap between the PI and PID controllers, for the PID obtaining better performance, lower IAE↓, for higher values of TV(u)↑ and TV(y)↑.

The system's time behavior in Figure 26 shows no significant differences for changes in L as R remained fixed. But it does confirm that the PID obtains a faster response than the PI, just as observed in Figure 24.

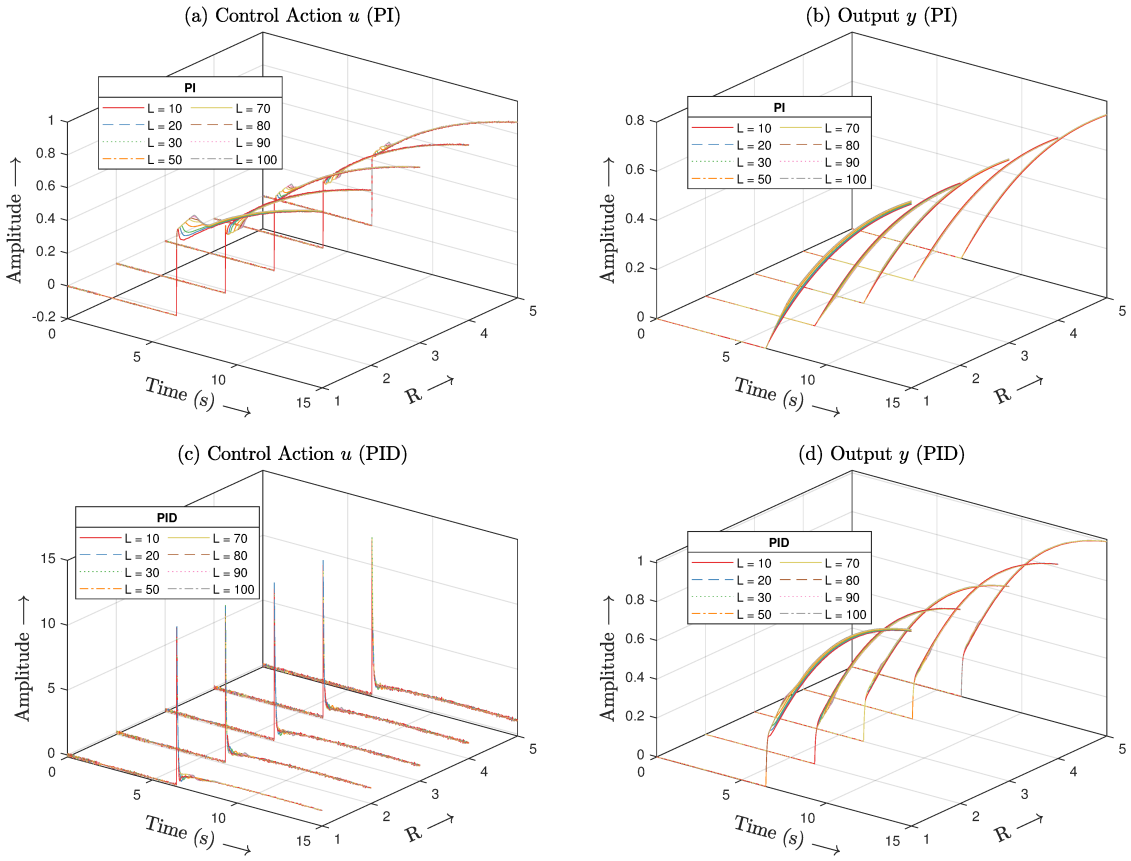


Figure 26 – Closed-loop performance of the balanced dynamics, for a unit step input, in the presence of measurement noise with SNR = 0.01, $N = 200$, L varying from 10 to 100 for each L, and $T_s = 0.01$ s ($\mathbf{R}$ fixed and $\mathbf{L}$ varying).

4.1.4.3 ***Delay-dominated dynamics***

The simulation results for the process P_D of delay-dominated dynamics are shown in Figure 27, 28, 29 and 30. The results for the absence of measurement noise were not significant and therefore are not included in Figure 27. Meanwhile, in the presence of measurement noise, Figure 27 (c)-(d), the patterns are similar to those obtained for process P_B of balanced dynamics, but with PI and PID controller's performances closer. As the noise attenuation degree decreases, lower $L \downarrow$ and higher $R \uparrow$, the two curves converge to ≈ 2.1 for PI and ≈ 1.95 for the PID. Also, the IAE values obtained are much lower than those of process P_B of balanced dynamics and slightly higher than those of P_L of lag-dominated dynamics.

The time behavior for the system output y in Figure 28 shows a similar performance for PI and PID. Likewise process P_B of balanced dynamics, no attenuation degree, i.e., combinations of L and R, destabilized the closed-loop system. This suggests that this type of process dynamics is less sensitive to changes in the noise filter attenuation.

The exact same results were obtained for the scenario with varying L for fixed values of R, comparing Figure 29 with Figure 27. The same can be concluded from the time

behavior of the closed-loop system for this case, given in Figure 30, compared to Figure 28.

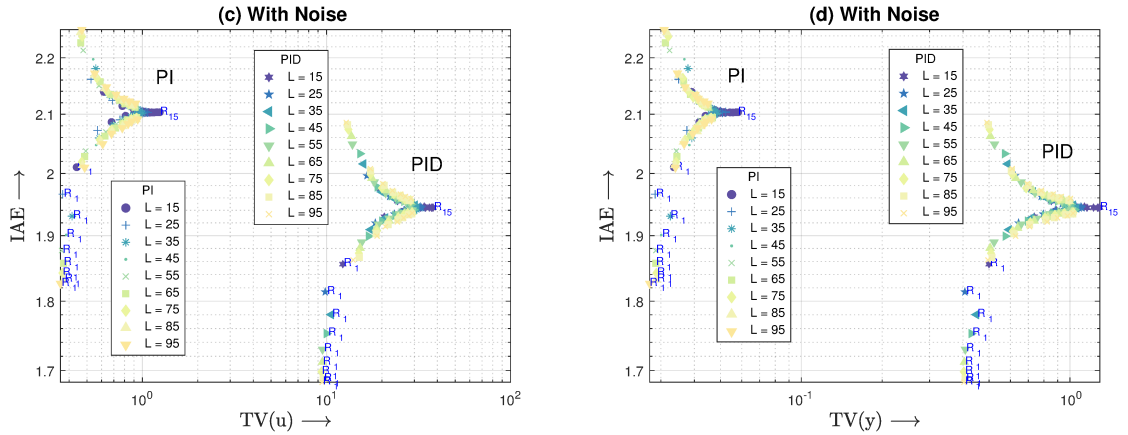


Figure 27 – Performance (IAE) vs. Measurement noise (TV) of the control signal u and process variable y for a delay-dominated dynamics, unit step input, SNR = 0.01, $N = 200$, R varying from 1 to 15 for each L, and $T_s = 0.01$ s ($\mathbf{L}$ fixed and $\mathbf{R}$ varying).

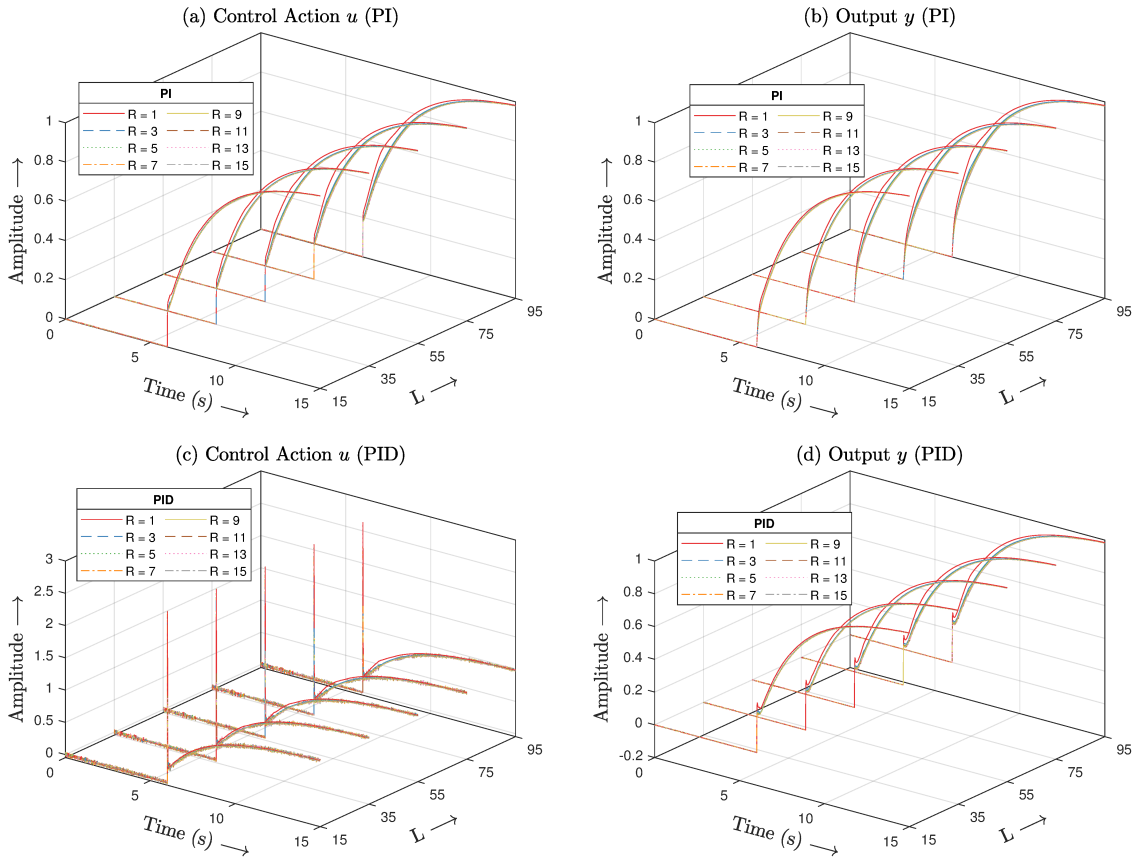


Figure 28 – Closed-loop performance of the delay-dominated dynamics, for a unit step input, in the presence of measurement noise with SNR = 0.01, $N = 200$, R varying from 1 to 15 for each L, and $T_s = 0.01$ s ($\mathbf{L}$ fixed and $\mathbf{R}$ varying).

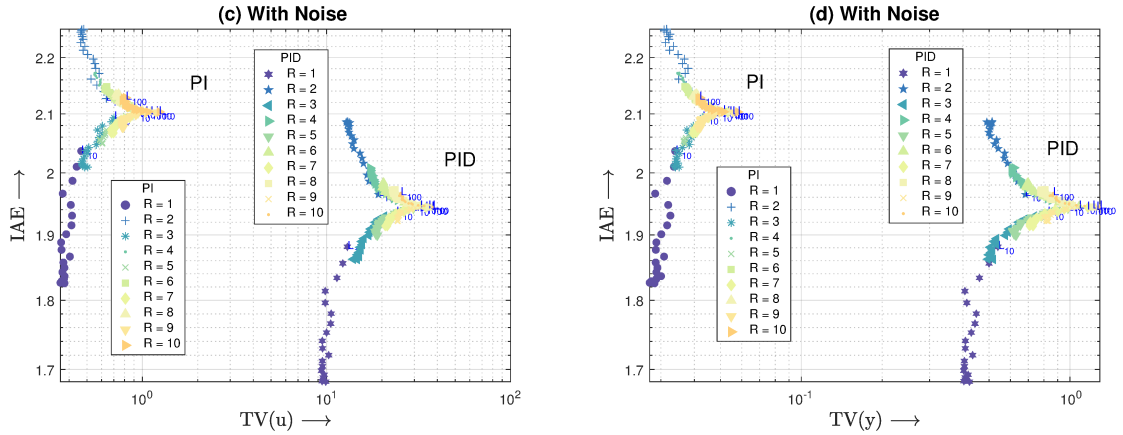


Figure 29 – Performance (IAE) vs. Measurement noise (TV) of the control signal u and process variable y for a delay-dominated dynamics, unit step input, SNR = 0.01, $N = 200$, L varying from 10 to 100 for each R, and $T_s = 0.01$ s ($\mathbf{R}$ fixed and $\mathbf{L}$ varying).

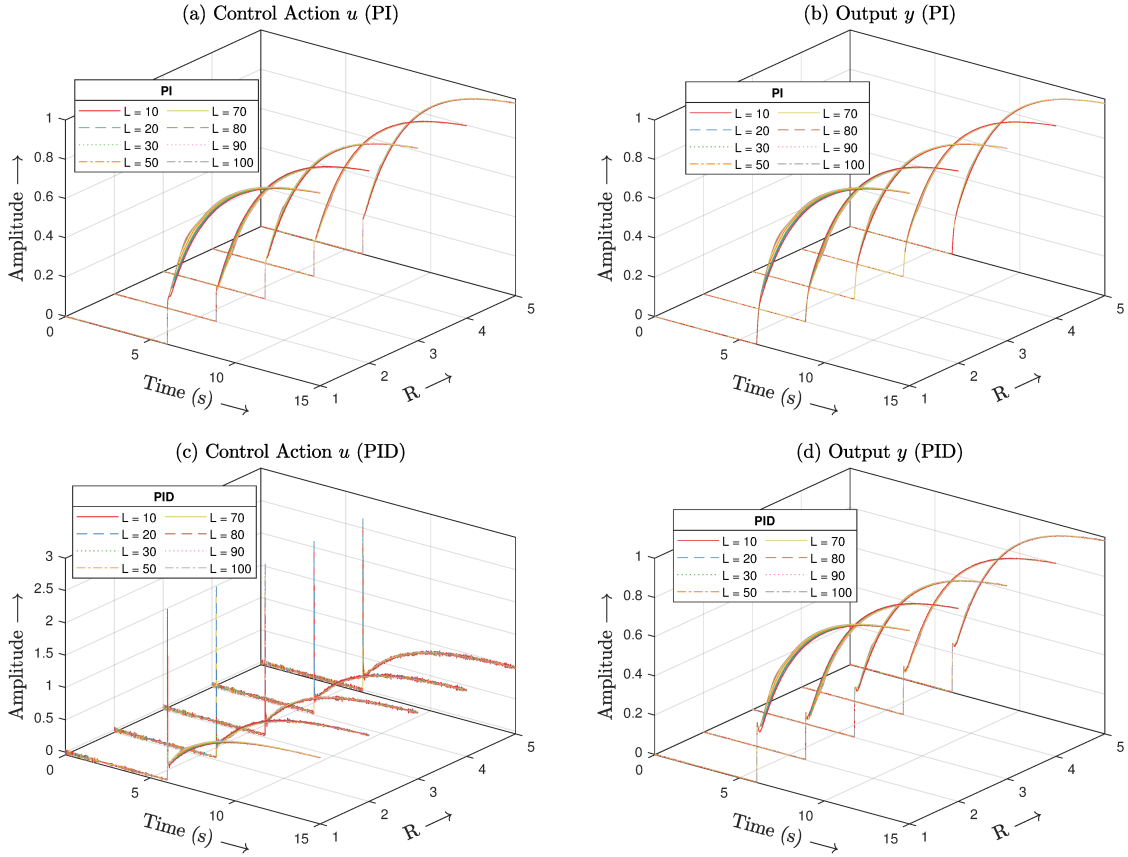


Figure 30 – Closed-loop performance of the delay-dominated dynamics, for a unit step input, in the presence of measurement noise with SNR = 0.01, $N = 200$, L varying from 10 to 100 for each L, and $T_s = 0.01$ s ($\mathbf{R}$ fixed and $\mathbf{L}$ varying).

4.2 Trajectory Filter SSA

The proposed method consists of replacing the FIR filter of the trajectory generating scheme in Figure 3 by the real-time version of the non-parametric SSA filter, the CSSA,

according to Figure 31.

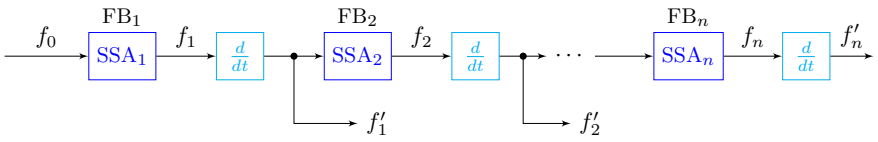


Figure 31 – Block diagram of a non-parametric trajectory filter of cascaded CSSA FIR FB.

Instead of a single parameter, as was T_i for the moving average filter, the CSSA requires three parameters to be tuned: (i) the sliding window length N; (ii) the embedding dimension L; and (iii) the number of components for reconstructing the time series R. Consequently, to investigate the response of the proposed trajectory filter, the experiments sought to vary one of the parameters, L or R, while the other remained unchanged.

4.2.1 Experimental Results

Figure 32, 33, and 34 show the results of the CSSA and the parametric trajectory filters applied to the following signals: pulse, ramp and sawtooth, respectively. These signals and their filtered versions for the CSSA and the FIR filter are shown in Figure 32-34 (a). While its first and second filtered derivatives are shown in Figure 32-34 (b) and (c), respectively.

The beginning of the simulations in which the filter was not yet active but only accumulating samples to The beginning sections of the simulations are hidden, given that the filter was not yet active but only accumulating samples to reach the minimum sliding window length N. The simulations in Figure 32 and 33 were performed for $N = 150$ samples, with sampling period $T_s = 0.01\,\text{s}$, signal's decomposition into $L = 100$ components, and reconstructing with only the first, $R = 1$. While in Figure 34, were used $N = 50$, $T_s = 0.1\,\text{s}$, $L = 40$. For the parametric filter were chosen $a_0 = 10^2$, $a_1 = 120$ and $a_2 = 21$.

The results in Figure 32, 33 and 34 show that the parametric and non-parametric trajectory filters obtained similar results, with the non-parametric resulting in signals with a slightly larger amplitude. Nonetheless, they returned bounded derivatives from an initial discontinuous input signal, at least for Figure 32 and 34. The results in Figure 33 show that if the input is continuous, filtering does not distort the signal derivatives significantly.

To further analyze and compare the non-parametric trajectory filtering response, the values of L and R were varied. The results are shown in Figure 35, for L in $[10\!:\!5\!:\!45]$ for each value of $R = [1, 2, 3, 4]$, and Figure 36, for R varying in $[1, 2, \ldots, 10]$ for each value of $L = [10, 20, 30, 40]$. The simulation used $N = 50$, $T_s = 0.1\,\text{s}$, $a_0 = 10^3$, $a_1 = 10^2$ and $a_2 = 30$.

In Figure 35 the best results were obtained for $R = 1$, the lowest possible value. In this case, by reconstructing the signal with its most important component, the method

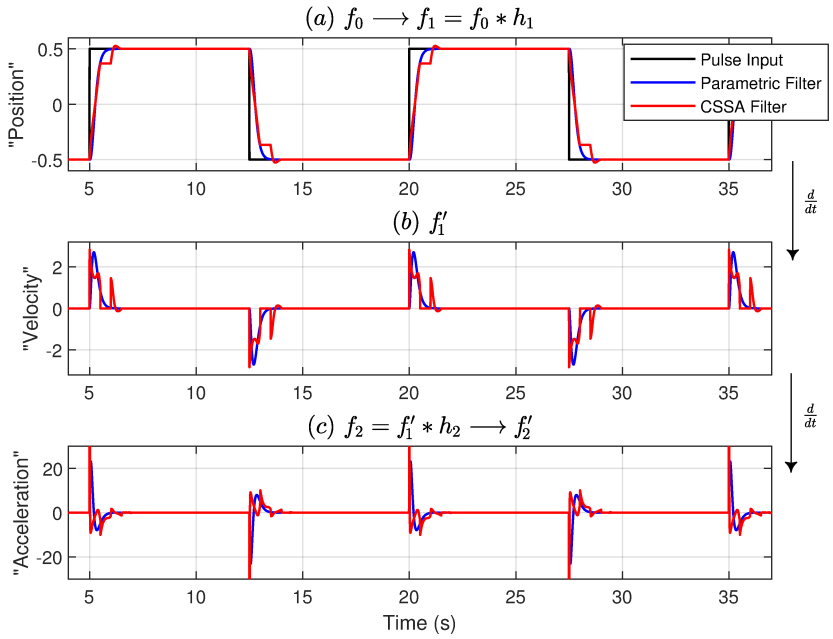


Figure 32 – CSSA and a parametric trajectory filter for the first two derivatives of a discontinuous pulse signal: $N = 150$, $T_s = 0.01$ s, $L = 100$, $R = 1$, $a_0 = 10^2$, $a_1 = 120$ and $a_2 = 21$.

provides a ramp-like signal that was faster than the ramp obtained by the parametric filter. The exception was for $L = 25$. All the other values of $R > 1$, regardless of the values of L, provided faster signals, in comparison with the parametric filter. However, it resulted in a distorted signal with some overshoot or undershoot. Once again, the case for $L = 25$

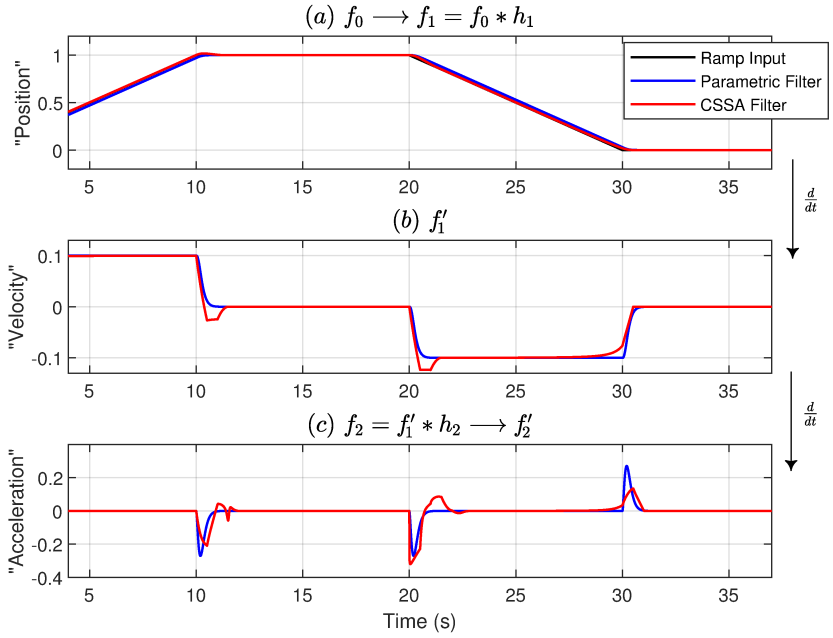


Figure 33 – CSSA and a parametric trajectory filter for the first two derivatives of a continuous ramp signal: $N = 150$, $T_s = 0.01$ s, $L = 100$, $R = 1$, $a_0 = 10^2$, $a_1 = 120$ and $a_2 = 21$.

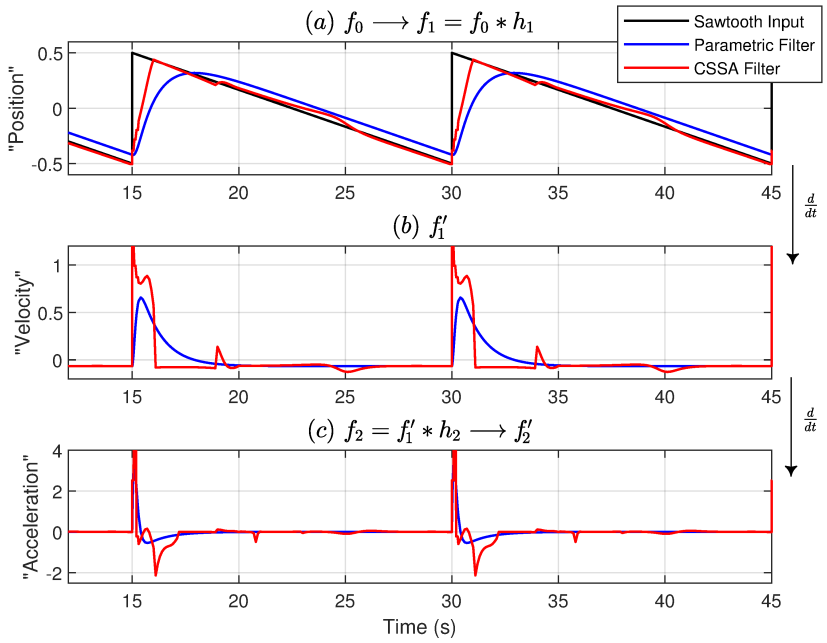


Figure 34 – CSSA and a parametric trajectory filter for the first two derivatives of a continuous sawtooth signal: $N = 50$, $T_s = 0.1$ s, $L = 40$, $R = 1$, $a_0 = 10^2$, $a_1 = 120$ and $a_2 = 21$.

exhibited the worst performance.

The fastest response is verified for $R = 1$ and $L = 45$, both in its extreme limits since $N = 50$ is an upper bound for L and R can not be lower than 1. Additionally, in all cases, the pair of values of $L = (10, 40)$, $(15, 35)$ and $(20, 30)$ showed a similar response.

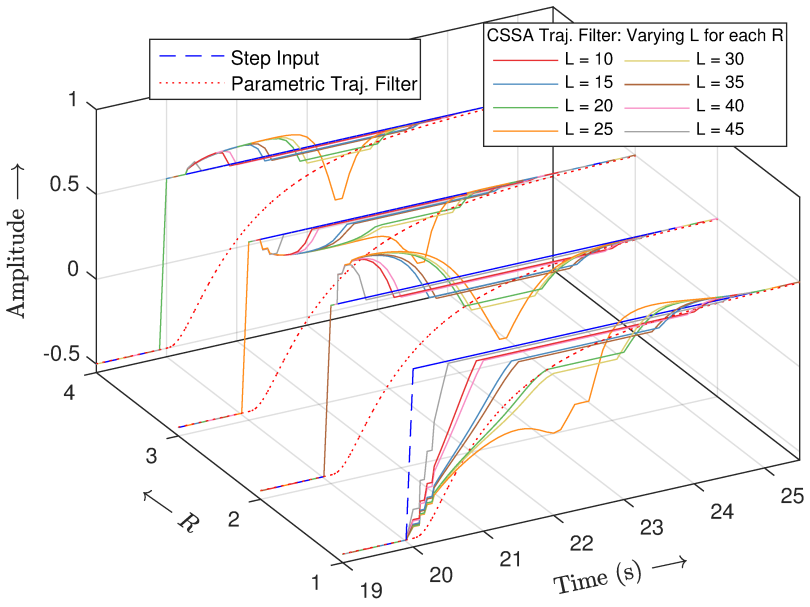


Figure 35 – CSSA trajectory filter for a discontinuous step signal for varying $\mathbf{L}$ for each value of $\mathbf{R}$: $N = 50$, $T_s = 0.1$ s, $L = [10{:}5{:}45]$, $R = [1, 2, 3, 4]$, $a_0 = 10^3$, $a_1 = 10^2$ and $a_2 = 30$.

Indicating that lower values of L could achieve a similar smoothness degree. But with a lower computational cost, given that the decomposition occurs in a lower dimension.

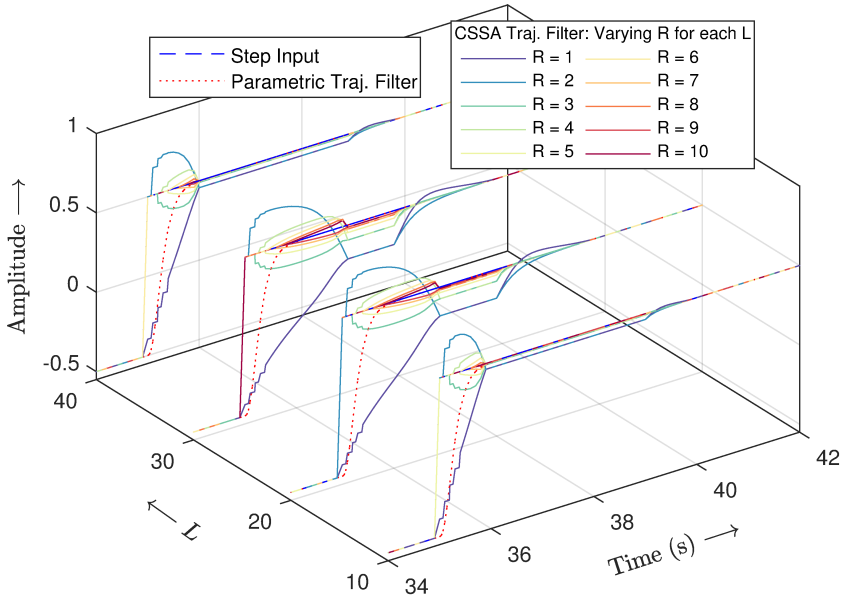


Figure 36 – CSSA trajectory filter for a discontinuous step signal for varying $\mathbf{R}$ for each value of $\mathbf{L}$: $N = 50$, $T_s = 0.1\,\text{s}$, $L = [10, 20, 30, 40]$, $R = [1, 2, \ldots, 10]$, $a_0 = 10^3$, $a_1 = 10^2$ and $a_2 = 30$.

Similar results were verified considering the inverse process, i.e., by varying R for each constant value of L, as shown in Figure 36. For any value of L considered, only for $R = 1$, the non-parametric trajectory filter was able to smooth the step signal to some degree. The reconstruction of the original signal with any other component besides the first and most important one does not result in a smoother signal, however, it still results in bounded and fast changes, approximating the step signal.

CHAPTER

5

CONCLUSIONS

5.1 Conclusions

The present work introduced preliminary concepts about PID control, controller tuning methods and the controller design with measurement noise attenuation with an appropriate noise filter, and a few related works on this issue. To motivate the use of a robust but straightforward controller synthesis, the AMIGO tuning method. And the need to further investigate the use of different filter structures and concepts to address the problem of including measurement noise attenuation requirements in the design phase of controller tuning methods.

Following this, the SSA main aspects were described, and related works in different areas of science that used the method were presented. Motivating aspects regarding the design of trajectory generating filters for non-linear control laws were also discussed, with related works on this issue.

The basic SSA methodology was then described, followed by its real-time version, the CSSA method. The frequency-domain interpretation of the SSA as a linear filtering process was also provided. The methodological aspects ended by discussing two examples on the time and frequency-domain analysis of time series with the SSA.

In this context, this study aimed to investigate the effects of using a non-parametric noise filter based on the SSA method in PID control loops, and the use of the SSA in a novel trajectory filter approach, suitable for nonlinear control methods.

Regarding the first, an exploratory analysis was conducted to investigate the use of a non-parametric SSA based noise filter in linear control with PID controllers. The numerical simulations consisted of varying the filter's attenuation degree for processes with balanced, lag, and delay-dominated dynamics, for PI and PID controllers, both in the presence and absence of measurement noise. The trade-off between performance and measurement

noise attenuation was analyzed through the IAE and TV of the control action and the process variable, using the CSSA method.

In the absence of measurement noise, no filtering achieved the best performances for both PI and PID controllers, regardless of the process dynamics. Since there is no noise, the SSA filters relevant components of the error signal, causing the control action and output performance to deteriorate, with an unnecessary increase in variability.

Meanwhile, the trade-off between better performance, lower IAE, and greater noise attenuation, lower TV, becomes clearly noticeable in the presence of measurement noise. For lag-dominated dynamics, a higher attenuation degree resulted in less signal variability, to a certain extent, at the expense of worse performance.

On the other hand, for the balanced and delay-dominated dynamics, for some values of L and R, a higher attenuation degree simultaneously improved both the performance and TV. In addition, the PID achieved better performance than the PI for all processes investigated, regardless of the presence or absence of measurement noise in the closed-loop system.

The largest discrepancy between PI and PID was obtained for the balanced dynamics, followed by the delay, and the lag-dominated dynamics. Furthermore, the process with balanced dynamics showed greater sensitivity to changes in the filter's attenuation degree. Resulting in an unstable closed-loop system for the PI and marginally stable for the PID in extreme scenarios for the values of L and R.

In contrast, the delay and lag-dominated dynamics showed robustness to changes in L and R, resulting in stable systems with fairly similar response curves. The results showed that the proposed method based on the use of the CSSA as a non-parametric noise filter can be successfully applied in linear control application.

The second analysis was the use of the CSSA in a novel trajectory filter generator. The exploratory results showed that the proposed non-parametric CSSA filter could be employed in an online trajectory generating scheme. The simulations returned bounded derivatives from initially discontinuous input signals such as steps, pulses and ramps.

The proposed filter followed the same principle applied to a cascade of FIR filters, which is the successive filtering of the input signal and its derivatives to smooth the input signal and get bounded derivatives, allowing the use of non-linear control laws such as feedback linearization and backstepping. These results are encouraging, since the frequency analysis for nonlinear system is a far more complex task than the linear counterpart.

5.2 Future Works

Given the plausibility of using the SSA as a noise filter in PID control loops and as a smoother filter in trajectory filter generators, the following suggestions to further extends this research are given:

- Expand the analysis to a larger and more significant batch of processes dynamics;
- Test the method with other controllers PID tuning methods;
- Extends the proposal to other linear and nonlinear controllers (more complex than a PID);
- Apply or propose a recursive version of the SSA, to reduce the computational cost related to the SVD step in CSSA; and
- Validate the methodology in a real control problem.

An open avenue for future research is an automatic algorithm for component selection in the grouping step, which can balance optimal or sub-optimal trade-offs among the controller's design requirements. That would also rely on graphical information generated by the filter. Another interesting issue is to devise methods to automatically choose values for the filter parameters and make them adaptive to changes in the time series structures.

CHAPTER

6

PUBLICATIONS

The findings described in this master thesis were published in three conference papers and are listed as follows:

1. SILVA, E. A. da; MOZELLI, L. A.; LELES, M. C. R. Non-parametric Noise Filtering in PID Control Loops using Singular Spectrum Analysis. In: **Procedings do XXIV Congresso Brasileiro de Automática**. SBA Sociedade Brasileira de Automática, 2022. (CBA2022). ISSN 2525-8311. Available: <https://www.sba.org.br/open_journal_systems/index.php/cba/article/view/3749>.

2. SILVA, E. A. da; MOZELLI, L. A.; CAMPOS, V. C. da S. On-line Generation of Trajectories Using Singular Spectrum Analysis: A Non-parametric Approach. In: **Procedings do XXIV Congresso Brasileiro de Automática**. SBA Sociedade Brasileira de Automática, 2022. (CBA2022). ISSN 2525-8311. Available: <https://www.sba.org.br/open_journal_systems/index.php/cba/article/view/3702>.

3. SILVA, E. A.; MOZELLI, L. A.; LELES, M. C. R.; CAMPOS, V. C. S.; PALAZZO, G. A study on Non-parametric Filtering in Linear and Nonlinear Control Loops using the Singular Spectrum Analysis. In: **2023 IEEE International Systems Conference (SysCon)**. IEEE, 2023. v. 54, p. 1–8. Available: <https://ieeexplore.ieee.org/abstract/document/10131113>.

BIBLIOGRAPHY

ALEXANDROV, T. A Method of Trend Extraction Using Singular Spectrum Analysis. **REVSTAT–Statistical Journal**, v. 7, n. 1, p. 1–22, Apr. 2009. Available: <https://www.ine.pt/revstat/pdf/rs090101.pdf>. Citations on pages 31 and 32.

ALONSO, F. J.; CASTILLO, J. M. D.; PINTADO, P. Application of Singular Spectrum Analysis to the Smoothing of Raw Kinematic Signals. **Journal of Biomechanics**, Elsevier BV, v. 38, n. 5, p. 1085–1092, may 2005. Available: <https://www.sciencedirect.com/science/article/pii/S0021929004002702>. Citation on page 32.

ÅSTRÖM, K. J.; HÄGGLUND, T. Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins. **Automatica**, Elsevier BV, v. 20, n. 5, p. 645–651, sep 1984. Available: <https://www.sciencedirect.com/science/article/pii/0005109884900141>. Citation on page 91.

____. **PID Controllers: Theory, Design, and Tuning**. [S.l.]: ISA The Instrumentation, Systems, and Automation Society, 1995. Citations on pages 20, 80, 81, 82, 83, 86, 87, and 88.

____. The Future of PID Control. **Control Engineering Practice**, Elsevier, v. 9, n. 11, p. 1163–1175, nov 2001. Available: <https://www.sciencedirect.com/science/article/pii/S0967066101000624>. Citations on pages 17 and 22.

____. Revisiting the Ziegler-Nichols Step Response Method for PID Control. **Journal of Process Control**, Elsevier BV, v. 14, n. 6, p. 635–650, sep 2004. Available: <https://www.sciencedirect.com/science/article/pii/S0959152404000034>. Citation on page 90.

____. **Advanced PID Control**. Research Triangle Park: ISA-The Instrumentation, Systems, and Automation Society, 2006. Citations on pages 20, 21, 22, 23, 24, 80, 81, 82, 83, 84, 86, 87, 88, 89, and 90.

ÅSTRÖM, K. J.; MURRAY, R. M. **Feedback systems: an introduction for scientists and engineers**. Princeton university press, 2021. Available: <http://www.cds.caltech.edu/~murray/books/AM08/pdf/am08-complete_22Feb09.pdf>. Citations on pages 21, 80, 81, 82, and 86.

ÅSTRÖM, K. J.; PANAGOPOULOS, H.; HÄGGLUND, T. Design of PI Controllers based on Non-Convex Optimization. **Automatica**, Elsevier BV, v. 34, n. 5, p. 585–601, may 1998. Available: <https://www.sciencedirect.com/science/article/pii/S0005109898000119>. Citations on pages 89 and 90.

ÅSTRÖM, K. J.; WITTENMARK, B. **Computer-Controlled Systems: Theory and Design**. [S.l.]: Courier Corporation, 2013. Citation on page 84.

BAUER, M.; HORCH, A.; XIE, L.; JELALI, M.; THORNHILL, N. The Current State of Control Loop Performance Monitoring - A Survey of Application in Industry. **Journal of Process Control**, Elsevier BV, v. 38, p. 1–10, feb 2016. Available: <https://www.sciencedirect.com/science/article/pii/S0959152415002127>. Citation on page 21.

BENNETT, S. Development of the PID Controller. **IEEE Control Systems**, Institute of Electrical and Electronics Engineers (IEEE), v. 13, n. 6, p. 58–62, dec 1993. Available: <https://ieeexplore.ieee.org/abstract/document/248006>. Citation on page 21.

____. A Brief History of Automatic Control. **IEEE Control Systems**, Institute of Electrical and Electronics Engineers (IEEE), v. 16, n. 3, p. 17–25, jun 1996. Available: <https://ieeexplore.ieee.org/abstract/document/506394>. Citation on page 21.

____. The Past of PID Controllers. **Annual Reviews in Control**, Elsevier, v. 25, p. 43–53, jan 2001. Available: <https://www.sciencedirect.com/science/article/pii/S1367578801000050>. Citation on page 21.

BESSET, P.; BÉARÉE, R. FIR Filter-based Online Jerk-constrained Trajectory Generation. **Control Engineering Practice**, Elsevier BV, v. 66, p. 169–180, sep 2017. Available: <https://www.sciencedirect.com/science/article/pii/S096706611730151X>. Citations on pages 30 and 31.

BIAGIOTTI, L.; MELCHIORRI, C. FIR filters for Online Trajectory Planning with Time- and Frequency-domain Specifications. **Control Engineering Practice**, Elsevier BV, v. 20, n. 12, p. 1385–1399, dec 2012. Available: <https://www.sciencedirect.com/science/article/pii/S0967066112001669>. Citations on pages 27, 29, 30, and 31.

CASTRUCCI, P. L.; BITTAR, A.; SALES, R. M. **Controle Automático**. Rio de Janeiro: Grupo Gen-LTC, 2011. Citations on pages 84 and 88.

CHEN, D.; SEBORG, D. E. PI/PID Controller Design Based on Direct Synthesis and Disturbance Rejection. **Industrial & Engineering Chemistry Research**, American Chemical Society (ACS), v. 41, n. 19, p. 4807–4822, aug 2002. Available: <https://pubs.acs.org/doi/pdf/10.1021/ie010756m>. Citations on pages 85 and 86.

CHEN, Q.; DAM, T. van; SNEEUW, N.; COLLILIEUX, X.; WEIGELT, M.; REBISCHUNG, P. Singular Spectrum Analysis for Modeling Seasonal Signals from GPS Time Series. **Journal of Geodynamics**, Elsevier BV, v. 72, p. 25–35, dec 2013. Available: <https://www.sciencedirect.com/science/article/pii/S0264370713000902>. Citation on page 32.

CHIEN, K. L.; HRONES, J. A.; RESWICK, J. B. On the Automatic Control of Generalized Passive Systems. **Journal of Fluids Engineering**, ASME International, v. 74, n. 2, p. 175–183, feb 1952. Available: <https://asmedigitalcollection.asme.org/fluidsengineering/article/74/2/175/1144365/On-the-Automatic-Control-of-Generalized-Passive>. Citation on page 82.

COHEN, G. H.; COON, G. A. Theoretical Consideration of Retarded Control. **Journal of Fluids Engineering**, ASME International, v. 75, n. 5, p. 827–834, jul 1953. Available: <https://asmedigitalcollection.asme.org/fluidsengineering/article/75/5/827/1145041/Theoretical-Consideration-of-Retarded-Control>. Citation on page 83.

COVID-19. **Covid-19 Brazil Datasets Painel**. 2022. Accessed: 2022-10-02. Available: <https://covid.saude.gov.br/>. Citation on page 42.

DAHLIN, E. B. Designing and Tuning Digital Controllers. **Instruments and Control systems**, v. 41, n. 6, p. 77–83, 1968. Citation on page 86.

ENDER, D. B. Process control performance: Not as good as you think. **Control Engineering**, v. 40, n. 10, p. 180–190, 1993. Citation on page 22.

FARRELL, J. A.; POLYCARPOU, M. M. **Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches**. [S.l.]: John Wiley & Sons, 2006. Citation on page 27.

FOLEY, M. W.; JULIEN, R. H.; COPELAND, B. R. A Comparison of PID Controller Tuning Methods. **The Canadian Journal of Chemical Engineering**, Wiley, v. 83, n. 4, p. 712–722, may 2008. Available: <https://onlinelibrary.wiley.com/doi/epdf/10.1002/cjce.5450830412>. Citation on page 90.

GARPINGER, O.; HÄGGLUND, T.; ÅSTRÖM, K. J. Criteria and Trade-offs in PID Design. **IFAC Proceedings Volumes**, Elsevier BV, v. 45, n. 3, p. 47–52, 2012. Available: <https://www.sciencedirect.com/science/article/pii/S1474667016309983>. Citation on page 23.

GERELLI, O.; BIANCO, C. G. L. A discrete-time filter for the on-line generation of trajectories with bounded velocity, acceleration, and jerk. In: **2010 IEEE International Conference on Robotics and Automation**. IEEE, 2010. p. 3989–3994. Available: <https://ieeexplore.ieee.org/abstract/document/5509712>. Citations on pages 29 and 30.

GOLYANDINA, N. On the choice of parameters in singular spectrum analysis and related subspace-based methods. **arXiv preprint arXiv:1005.4374**, 2010. Citation on page 44.

GOLYANDINA, N.; KOROBEYNIKOV, A. Basic Singular Spectrum Analysis and Forecasting with R. **Computational Statistics & Data Analysis**, Elsevier BV, v. 71, p. 934–954, mar 2014. Available: <https://www.sciencedirect.com/science/article/pii/S0167947313001394>. Citation on page 39.

GOLYANDINA, N.; NEKRUTKIN, V.; ZHIGLJAVSKY, A. A. **Analysis of Time Series Structure: SSA and Related Techniques**. [S.l.]: CRC press, 2001. Citation on page 31.

GOLYANDINA, N.; ZHIGLJAVSKY, A.; GOLYANDINA, N.; ZHIGLJAVSKY, A. SSA for Forecasting, Interpolation, Filtration and Estimation. In: **Singular Spectrum Analysis for Time Series**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 71–119. Available: <https://doi.org/10.1007/978-3-642-34913-3_3>. Citations on pages 31, 39, and 44.

HAAVISTO, O. Detection and Analysis of Oscillations in a Mineral Flotation Circuit. **Control Engineering Practice**, Elsevier BV, v. 18, n. 1, p. 23–30, jan 2010. Available: <https://www.sciencedirect.com/science/article/pii/S096706610900152X>. Citations on pages 32 and 33.

HÄGGLUND, T. Automatic Detection of Sluggish Control Loops. **Control Engineering Practice**, Elsevier BV, v. 7, n. 12, p. 1505–1511, dec 1999. Available: <https://www.sciencedirect.com/science/article/pii/S0967066199001161>. Citation on page 22.

____. Industrial Applications of Automatic Performance Monitoring Tools. **IFAC Proceedings Volumes**, Elsevier BV, v. 35, n. 1, p. 381–386, 2002. Available: <https://www.sciencedirect.com/science/article/pii/S1474667015400382>. Citation on page 22.

____. Signal Filtering in PID Control. **IFAC Proceedings Volumes**, Elsevier BV, v. 45, n. 3, p. 1–10, 2012. Available: <https://www.sciencedirect.com/science/article/pii/S1474667016309922>. Citation on page 22.

____. A Unified Discussion on Signal Filtering in PID Control. **Control Engineering Practice**, Elsevier BV, v. 21, n. 8, p. 994–1006, aug 2013. Available: <https://www.sciencedirect.com/science/article/pii/S096706611300066X>. Citation on page 23.

HÄGGLUND, T.; ÅSTRÖM, K. J. Revisiting the Ziegler-Nichols Tuning Rules for PI Control. **Asian Journal of Control**, Wiley, v. 4, n. 4, p. 364–380, oct 2002. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1934-6093.2002.tb00076.x>. Citation on page 90.

____. Revisiting the Ziegler-Nichols Tuning Rules for PI Control - Part II: The Frequency Response Method. **Asian Journal of Control**, Wiley, v. 6, n. 4, p. 469–482, oct 2004. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1934-6093.2004.tb00368.x>. Citation on page 90.

HARRIS, T. J.; YUAN, H. Filtering and Frequency Interpretations of Singular Spectrum Analysis. **Physica D: Nonlinear Phenomena**, Elsevier BV, v. 239, n. 20-22, p. 1958–1967, oct 2010. Available: <https://www.sciencedirect.com/science/article/pii/S0167278910002150>. Citation on page 46.

HASSANI, H. Singular Spectrum Analysis: Methodology and Comparison. **Journal of Data Science**, Cardiff University and Central Bank of the Islamic Republic of Iran, v. 5, n. 2, p. 239–257, Apr. 2007. Available: <https://mpra.ub.uni-muenchen.de/4991/>. Citations on pages 31 and 32.

HASSANI, H.; DIONISIO, A.; GHODSI, M. The Effect of Noise Reduction in Measuring the Linear and Nonlinear Dependency of Financial Markets. **Nonlinear Analysis: Real World Applications**, Elsevier BV, v. 11, n. 1, p. 492–502, feb 2010. Available: <https://www.sciencedirect.com/science/article/pii/S1468121809000029>. Citations on pages 32 and 33.

HASSANI, H.; THOMAKOS, D. A Review on Singular Spectrum Analysis for Economic and Financial Time Series. **Statistics and Its Interface**, International Press of Boston, v. 3, n. 3, p. 377–397, 2010. Available: <https://www.intlpress.com/site/pub/files/_fulltext/journals/sii/2010/0003/0003/SII-2010-0003-0003-a011.pdf>. Citation on page 32.

HO, W. K.; GAN, O. P.; TAY, E. B.; ANG, E. L. Performance and Gain and Phase Margins of Well-known PID Tuning Formulas. **IEEE Transactions on Control Systems Technology**, Institute of Electrical and Electronics Engineers (IEEE), v. 4, n. 4, p. 473–477, jul 1996. Available: <https://ieeexplore.ieee.org/abstract/document/508897>. Citation on page 91.

HUBA, M. Filter Choice for an Effective Measurement Noise Attenuation in PI and PID Controllers. In: **2015 IEEE International Conference on Mechatronics (ICM)**. IEEE, 2015. p. 46–51. Available: <https://ieeexplore.ieee.org/abstract/document/7083946>. Citations on pages 23, 24, and 52.

ISAKSSON, A. J.; GRAEBE, S. F. Derivative Filter is an Integral Part of PID Design. **IEE Proceedings - Control Theory and Applications**, Institution of Engineering and Technology (IET), v. 149, n. 1, p. 41–45, jan 2002. Available: <https://digital-library.theiet.org/content/journals/10.1049/ip-cta_20020111>. Citation on page 23.

JEON, J. W.; HA, Y. Y. A Generalized Approach for the Acceleration and Deceleration of Industrial Robots and CNC Machine Tools. **IEEE Transactions on Industrial Electronics**, Institute of Electrical and Electronics Engineers (IEEE), v. 47, n. 1, p. 133–139, 2000. Available: <https://ieeexplore.ieee.org/abstract/document/824135>. Citation on page 30.

KIM, D.-I.; JEON, J. W.; KIM, S. Software Acceleration/Deceleration Methods for Industrial Robots and CNC Machine Tools. **Mechatronics**, Elsevier BV, v. 4, n. 1, p. 37–53, feb 1994. Available: <https://www.sciencedirect.com/science/article/pii/0957415894900493>. Citations on pages 29 and 30.

KRISTIANSSON, B.; LENNARTSON, B. Evaluation and Simple Tuning of PID Controllers with High-Frequency Robustness. **Journal of Process Control**, Elsevier BV, v. 16, n. 2, p. 91–102, feb 2006. Available: <https://www.sciencedirect.com/science/article/pii/S0959152405000594>. Citations on pages 23 and 25.

KUME, K. Interpretation of Singular Spectrum Analysis as Complete Eigenfilter Decomposition. **Advances in Adaptive Data Analysis**, World Scientific Pub Co Pte Ltd, v. 04, n. 04, p. 1250023, oct 2012. Citation on page 46.

LEE, Y.; PARK, S.; LEE, M.; BROSILOW, C. PID Controller Tuning for Desired Closed-loop Responses for SI/SO Systems. **AIChE Journal**, Wiley, v. 44, n. 1, p. 106–115, jan 1998. Available: <https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690440112>. Citation on page 90.

LELES, M. C. R. **Análise espectral singular aplicada no desenvolvimento de estratégias de negociação para o mercado de capitais**. Phd Thesis (PhD Thesis) — Universidade Federal de Minas Gerais, Escola de Engenharia, 2017. Available: <https://ppgee.ufmg.br/defesas/279D.PDF>. Citation on page 44.

LELES, M. C. R.; CARDOSO, A. S. V.; MOREIRA, M. G.; GUIMARAES, H. N.; SILVA, C. M.; PITSILLIDES, A. Frequency-domain Characterization of Singular Spectrum Analysis Eigenvectors. In: **2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)**. IEEE, 2016. Available: <https://ieeexplore.ieee.org/abstract/document/7886003>. Citation on page 46.

LELES, M. C. R.; MOZELLI, L. A.; GUIMARAES, H. N. A New Trend-Following Indicator: Using SSA to Design Trading Rules. **Fluctuation and Noise Letters**, World Scientific Pub Co Pte Ltd, v. 16, n. 02, p. 1750016–1–16, mar 2017. Available: <https://www.worldscientific.com/doi/abs/10.1142/S021947751750016X>. Citation on page 40.

LU, Y.-S. Smooth Speed Control of Motor Drives with Asymptotic Disturbance Compensation. **Control Engineering Practice**, Elsevier BV, v. 16, n. 5, p. 597–608, may 2008. Available: <https://www.sciencedirect.com/science/article/pii/S0967066107001232>. Citations on pages 29 and 30.

MICIĆ, A. D.; MATAUŠEK, M. R. Optimization of PID Controller with Higher-order Noise Filter. **Journal of Process Control**, Elsevier BV, v. 24, n. 5, p. 694–700, may 2014. Available: <https://www.sciencedirect.com/science/article/pii/S0959152413002163>. Citation on page 26.

O'DWYER, A. **Handbook of PI and PID Controller Tuning Rules**. [S.l.]: World Scientific, 2009. Citation on page 80.

OLABI, A.; BÉARÉE, R.; GIBARU, O.; DAMAK, M. Feedrate Planning for Machining with Industrial Six-axis Robots. **Control Engineering Practice**, Elsevier BV, v. 18, n. 5, p. 471–482, may 2010. Available: <https://www.sciencedirect.com/science/article/pii/S0967066110000055>. Citation on page 30.

PHUONG, T. T.; OHISHI, K.; YOKOKURA, Y. Advanced Wideband Sensorless Force Control Based on Harmonic Estimation Integrated Singular Spectrum Analysis Based Disturbance Observer. In: **IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society**. IEEE, 2019. Available: <https://ieeexplore-ieee-org.ez27.periodicos.capes.gov.br/document/8926873>. Citations on pages 32 and 34.

____. Enhancement of Performance on Sensor-less Force Sensation Using Singular Spectrum Analysis Based Force Observers. In: **2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)**. IEEE, 2020. Available: <https://ieeexplore.ieee.org/abstract/document/9158645>. Citations on pages 32 and 35.

PHUONG, T. T.; OHISHI, K.; YOKOKURA, Y.; BO, T. X.; YABUKI, A. High Performance Load Acceleration Control Based on Singular Spectrum Analysis for Industrial Robot. In: **2016 IEEE International Power Electronics and Motion Control Conference (PEMC)**. IEEE, 2016. Available: <https://ieeexplore.ieee.org/abstract/document/7752100>. Citations on pages 32 and 34.

RIVERA, D. E.; MORARI, M.; SKOGESTAD, S. Internal Model Control: PID Controller Design. **Industrial & Engineering Chemistry Process Design and Development**, ACS Publications, v. 25, n. 1, p. 252–265, 1986. Available: <https://pubs.acs.org/doi/pdf/10.1021/i200032a041>. Citation on page 87.

SALGADO, D. R.; ALONSO, F. J. Tool Wear Detection in Turning Operations Using Singular Spectrum Analysis. **Journal of Materials Processing Technology**, Elsevier BV, v. 171, n. 3, p. 451–458, feb 2006. Available: <https://www.sciencedirect.com/science/article/pii/S0924013605008083>. Citations on pages 32 and 33.

SEBORG, D. E.; EDGAR, T. F.; MELLICHAMP, D. A.; III, F. J. D. **Process Dynamics and Control**. [S.l.]: John Wiley & Sons, 1989. Citations on pages 85 and 86.

SEGOVIA, V. R.; HÄGGLUND, T.; ÅSTRÖM, K. J. Noise Filtering in PI and PID Control. In: **2013 American Control Conference**. IEEE, 2013. p. 1763–1770. Available: <https://ieeexplore.ieee.org/abstract/document/6580091>. Citations on pages 22, 24, 25, and 51.

____. Measurement Noise Filtering for PID Controllers. **Journal of Process Control**, Elsevier BV, v. 24, n. 4, p. 299–313, apr 2014. Available: <https://www.sciencedirect.com/science/article/pii/S0959152414000419>. Citations on pages 22, 23, 24, 25, and 51.

____. Design of Measurement Noise Filters for PID Control. **IFAC Proceedings Volumes**, Elsevier BV, v. 47, n. 3, p. 8359–8364, 2014a. Available: <https://www.sciencedirect.com/science/article/pii/S1474667016429320>. Citations on pages 22, 23, 25, 87, 89, and 91.

____. Measurement Noise Filtering for Common PID Tuning Rules. **Control Engineering Practice**, Elsevier BV, v. 32, p. 43–63, nov 2014b. Available: <https://www.sciencedirect.com/science/article/pii/S0967066114001750>. Citations on pages 23 and 25.

SEKARA, T. B.; MATAUSEK, M. R. Optimization of PID Controller Based on Maximization of the Proportional Gain Under Constraints on Robustness and Sensitivity to Measurement Noise. **IEEE Transactions on Automatic Control**, Institute of Electrical and Electronics Engineers (IEEE), v. 54, n. 1, p. 184–189, jan 2009. Available: <https://ieeexplore.ieee.org/abstract/document/4749444>. Citations on pages 23 and 24.

ŠEKARA, T. B.; TRIFUNOVIĆ, M. B.; GOVEDARICA, V. Frequency Domain Design of a Complex Controller Under Constraints on Robustness and Sensitivity to Measurement Noise. **Electronics**, v. 15, n. 1, p. 40–44, Jun. 2011. Available: <http://els-journal.etf.unibl.org/journal/Vol15No1/xPaper_06.pdf>. Citation on page 26.

SILVA, E. A.; MOZELLI, L. A.; LELES, M. C. R.; CAMPOS, V. C. S.; PALAZZO, G. A study on Non-parametric Filtering in Linear and Nonlinear Control Loops using the Singular Spectrum Analysis. In: **2023 IEEE International Systems Conference (SysCon)**. IEEE, 2023. v. 54, p. 1–8. Available: <https://ieeexplore.ieee.org/abstract/document/10131113>. Citation on page 71.

SILVA, E. A. da; MOZELLI, L. A.; CAMPOS, V. C. da S. On-line Generation of Trajectories Using Singular Spectrum Analysis: A Non-parametric Approach. In: **Procedings do XXIV Congresso Brasileiro de Automática**. SBA Sociedade Brasileira de Automática, 2022. (CBA2022). ISSN 2525-8311. Available: <https://www.sba.org.br/open_journal_systems/index.php/cba/article/view/3702>. Citation on page 71.

SILVA, E. A. da; MOZELLI, L. A.; LELES, M. C. R. Non-parametric Noise Filtering in PID Control Loops using Singular Spectrum Analysis. In: **Procedings do XXIV Congresso Brasileiro de Automática**. SBA Sociedade Brasileira de Automática, 2022. (CBA2022). ISSN 2525-8311. Available: <https://www.sba.org.br/open_journal_systems/index.php/cba/article/view/3749>. Citation on page 71.

SKOGESTAD, S. Simple Analytic Rules for Model Reduction and PID Controller Tuning. **Journal of Process Control**, Elsevier BV, v. 13, n. 4, p. 291–309, jun 2003. Available: <https://www.sciencedirect.com/science/article/pii/S0959152402000628>. Citation on page 88.

TAN, W.; LIU, J.; CHEN, T.; MARQUEZ, H. J. Comparison of Some Well-known PID Tuning Formulas. **Computers & Chemical Engineering**, Elsevier BV, v. 30, n. 9, p. 1416–1423, jul 2006. Available: <https://www.sciencedirect.com/science/article/pii/S0098135406000718>. Citation on page 90.

TOMé, A. M.; MALAFAIA, D.; TEIXEIRA, A. R.; LANG, E. W. On the Use of Singular Spectrum Analysis. **arXiv preprint arXiv:1807.10679**, 2018. Available: <https://arxiv.org/abs/1807.10679>. Citations on pages 45 and 46.

VILANOVA, R.; VISIOLI, A. **PID Control in the Third Millennium: Lessons Learned and New Approaches**. Springer, 2012. ISBN 9781447124245. Available: <https://link.springer.com/book/10.1007/978-1-4471-2425-2>. Citations on pages 21 and 80.

WITRANT, E.; FRIDMAN, E.; SENAME, O.; DUGARD, L. **Recent results on time-delay systems: Analysis and control**. [S.l.]: Springer, 2016. Citation on page 21.

YU, C.-C. **Autotuning of PID Controllers: A Relay Feedback Approach**. Springer Science & Business Media, 2006. ISBN 978-1-4471-3638-5. Available: <https://link.springer.com/book/10.1007/978-1-4471-3636-1>. Citation on page 21.

ZANASI, R.; BIANCO, C. G. L.; TONIELLI, A. Nonlinear Filters for the Generation of Smooth Trajectories. **Automatica**, Elsevier BV, v. 36, n. 3, p. 439–448, mar 2000. Available: <https://www.sciencedirect.com/science/article/pii/S0005109899001648>. Citations on pages 29 and 30.

ZANASI, R.; MORSELLI, R. Discrete Minimum Time Tracking Problem for a Chain of Three Integrators with Bounded Input. **Automatica**, Elsevier BV, v. 39, n. 9, p. 1643–1649, sep 2003. Available: <https://www.sciencedirect.com/science/article/pii/S0005109803001699>. Citation on page 30.

ZHANG, H.; WANG, Z. Attitude Control and Sloshing Suppression for Liquid-Filled Spacecraft in the Presence of Sinusoidal Disturbance. **Journal of Sound and Vibration**, Elsevier BV, v. 383, p. 64–75, nov 2016. Available: <https://www.sciencedirect.com/science/article/pii/S0022460X16303893>. Citations on pages 32 and 34.

ZHENG, C.; SU, Y.; MüLLER, P. C. Simple Online Smooth Trajectory Generations for Industrial Systems. **Mechatronics**, Elsevier BV, v. 19, n. 4, p. 571–576, jun 2009. Available: <https://www.sciencedirect.com/science/article/pii/S0957415808001931>. Citations on pages 29 and 30.

ZHIGLJAVSKY, A. Singular Spectrum Analysis for Time Series: Introduction to This Special Issue. **Statistics and Its Interface**, International Press of Boston, v. 3, n. 3, p. 255–258, 2010. Available: <https://www.intlpress.com/site/pub/files/_fulltext/journals/sii/2010/0003/0003/SII-2010-0003-0003-a001.pdf>. Citation on page 31.

ZHUANG, M.; ATHERTON, D. P. Automatic Tuning of Optimum PID Controllers. **IEE Proceedings D Control Theory and Applications**, Institution of Engineering and Technology (IET), v. 140, n. 3, p. 216, 1993. Available: <https://digital-library.theiet.org/content/journals/10.1049/ip-d.1993.0030>. Citation on page 91.

ZIEGLER, J. G.; NICHOLS, N. B. Optimum Settings for Automatic Controllers. **Journal of Dynamic Systems, Measurement, and Control**, ASME International, v. 115, n. 2B, p. 220–222, jun 1942. Available: <https://po-files.ks3-cn-beijing.ksyun.com/5f7ebb8f1e085307a07c8cf6_fdad07ae78fc.pdf>. Citation on page 80.

APPENDIX

A

PID TUNING METHODS

The process of assigning values to a controller's parameters while complying with performance and robustness requirements is called controller tuning. Given its widespread use, numerous tuning methods are available for PID controllers (ÅSTRÖM; MURRAY, 2021). Åström and Hägglund (1995) and Åström and Hägglund (2006) are classic references on PID controllers and some of the main tuning methods. Vilanova and Visioli (2012) and O'dwyer (2009) present a comprehensive list of tuning methods for several process dynamics and controller structures.

In this section, some of the most used tuning methods are reviewed, starting from the simplest and most well-established like the Ziegler & Nichols (ZN) methods, to more complex methods that consider robustness in the design, such as the M-constrained Integral Gain Optimization (MIGO) and Approximated-MIGO (AMIGO) methods.

A.1 Ziegler & Nichols - Step Response Method

Proposed by Ziegler and Nichols (1942), this method consists of applying a step to the open-loop process to get an s-shaped response curve (ÅSTRÖM; HÄGGLUND, 2006), as shown in Figure 37. The parameters that characterize this response curve are the time delay τ_d, the time constant τ and a (ÅSTRÖM; MURRAY, 2021). They are obtained through the intersection of the tangent line, at the inflection point of the curve, with the time axis, to find τ_d, and the instant at which the tangent line intersects the response steady-state value, gives an estimate of τ (ÅSTRÖM; HÄGGLUND, 1995). The controller parameters for P, PI, and PID are calculated then as a function of these parameters, according to Table 3.

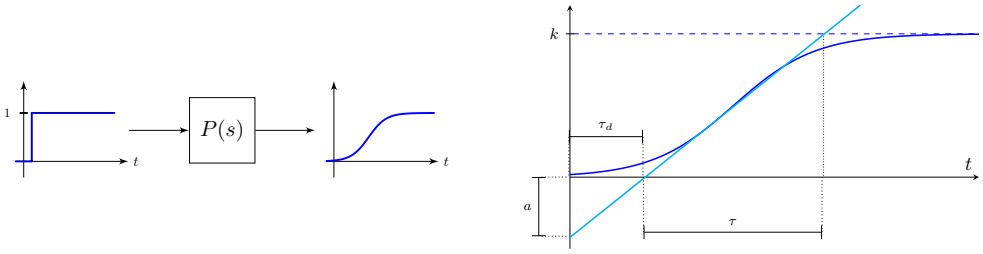


Figure 37 – (a) Open-loop step response and (b) process characteristics for the ZN step response method.

Table 3 – Controller parameters for ZN step response method.

Controller	ak_c	T_i/τ_d	T_d/τ_d
P	1	-	-
PI	0.9	3	-
PID	1.2	2	0.5

Source: Adapted from Åström and Hägglund (1995), Åström and Hägglund (2006).

A.2 Ziegler & Nichols - Frequency Response Method

The ZN frequency response method relies on the closed-loop characterization of the process dynamics (ÅSTRÖM; HÄGGLUND, 1995). The method consists of using only the proportional action k_c of the controller (i.e., $T_i = \infty$ and $T_d = 0$) (ÅSTRÖM; HÄGGLUND, 1995), Figure 38. After that, k_c is increased until the system response curve shows sustained oscillations (ÅSTRÖM; HÄGGLUND, 2006). At this instant, the critical gain k_{crit}, and the critical period T_{crit}, are obtained, as illustrated in Figure 39. Based on these parameters, simple formulas are use for the controller design (ÅSTRÖM; HÄGGLUND, 1995), which are given in Table 4.

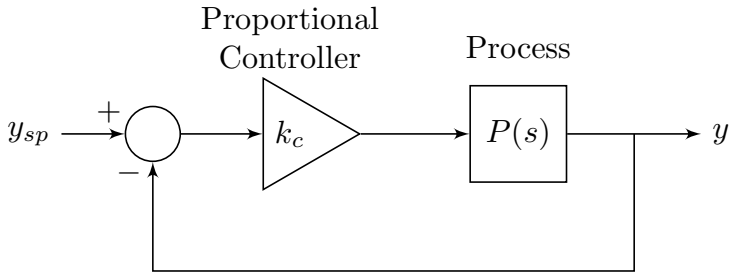


Figure 38 – Proportional control for the ZN frequency response method.

The ZN step and frequency response methods are classic methods that had a profound impact on the use of PID controllers and are still widely used by controller manufacturers and in industrial processes (ÅSTRÖM; HÄGGLUND, 2006). They rely on a few parameters obtained through the process response curve to devise simple controller tuning rules (ÅSTRÖM; HÄGGLUND, 2006; ÅSTRÖM; MURRAY, 2021). They were designed to

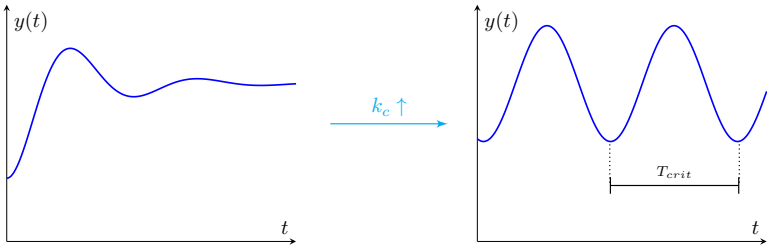


Figure 39 – ZN frequency response process: increase proportional gain k_c to obtained sustained oscillation.

Table 4 – Controller parameters for ZN frequency response method.

Controller	k_c/K_{crit}	T_i/T_{crit}	T_d/T_{crit}
P	0.5	-	-
PI	0.4	0.8	-
PID	0.6	0.5	0.125

Source: Adapted from Åström and Hägglund (1995), Åström and Hägglund (2006).

provide good rejection of load disturbances and have poor robustness, due to the use of a quarter-amplitude decay rate, leading to poorly damped and highly sensitive closed-loop systems (ÅSTRÖM; MURRAY, 2021).

A.3 Chien, Hrones e Reswick Tuning Method

The method of Chien, Hrones e Reswick (CHR), Chien, Hrones and Reswick (1952), is based on a First-Order Time Delay (FOTD) process model, Equation A.1, and uses the same information from the response curve, the time-delay and time constant, as the method of ZN (ÅSTRÖM; HÄGGLUND, 2006; ÅSTRÖM; HÄGGLUND, 1995). It aims to balance load disturbance rejection and set-point tracking capabilities, resulting in different formulas for the controller parameters, with smaller gains compared to the ZN methods (ÅSTRÖM; HÄGGLUND, 2006).

$$P(s) = \frac{k_p e^{-\tau_d s}}{\tau s + 1} \tag{A.1}$$

where k_p, τ_d and τ are the steady-state gain, the time delay, and the time constant, respectively.

Furthermore, the method accept requirements on the maximum allowed overshoot. The tuning rules are then given for two cases: (i) the fastest possible response without overshoot and (ii) the fastest possible response with a 20% maximum overshoot (ÅSTRÖM; HÄGGLUND, 2006). These rules are presented in Table 5 and Table 6.

It is a more robust method than ZN while still relying on just a few characteristics of the process dynamics (ÅSTRÖM; MURRAY, 2021).

Table 5 – Controller parameter for CHR load disturbance response method.

	No overshoot			20% overshoot		
Controller	ak_c	T_i/τ_d	T_d/τ_d	ak_c	T_i/τ_d	T_d/τ_d
P	0.3			0.7		
PI	0.6	4		0.7	2.3	
PID	0.95	2.4	0.42	1.2	2	0.42

Source: Adapted from Åström and Hägglund (1995), Åström and Hägglund (2006).

Table 6 – Controller parameter for CHR set-point method.

	No overshoot			20% overshoot		
Controller	ak_c	T_i/τ_d	T_d/τ_d	ak_c	T_i/τ_d	T_d/τ_d
P	0.3	-	-	0.7	-	-
PI	0.35	1.2	-	0.6	1	-
PID	0.6	1	0.5	0.95	1.4	0.47

Source: Adapted from Åström and Hägglund (1995), Åström and Hägglund (2006).

A.4 Cohen-Coon Tuning Method

The Cohen-Coon (CC), Cohen and Coon (1953), tuning method is more complex than the ZN and considers more information about the process dynamics. It was designed for load disturbance rejection and, similarly to the CHR method, approximates the process by a FOTD model (ÅSTRÖM; HÄGGLUND, 2006).

The method consists of placing the dominant closed-loop poles such that a one-fourth decay rate is guaranteed, as in ZN methods (ÅSTRÖM; HÄGGLUND, 2006). It does that by maximizing the proportional gain for P and PD controllers, and by maximizing the integral gain for PI and PID controllers, or equivalently, by minimizing the IE (ÅSTRÖM; HÄGGLUND, 2006). Table 7 show the tuning rules for this method.

Table 7 – Controller parameters for CC method.

Controller	P	PI	PD	PID
ak_c	$1+\frac{0.35\tau}{1-\tau}$	$0.9\left(1+\frac{0.092\tau}{1-\tau}\right)$	$1.24\left(1+\frac{0.13\tau}{1-\tau}\right)$	$1.35\left(1+\frac{0.18\tau}{1-\tau}\right)$
T_i/τ_d	-	$\frac{3.3-3\tau}{1+1.2\tau}$	-	$\frac{2.5-2\tau}{1-0.39\tau}$
T_d/τ_d	-	-	$\frac{0.27-0.36\tau}{1-0.87\tau}$	$\frac{0.37-0.37\tau}{1-0.81\tau}$

Source: Adapted from Åström and Hägglund (1995), Åström and Hägglund (2006).

A.5 Algebraic Pole Placement

The Algebraic Pole Placement (APP) method takes advantage of the fact that some characteristics of the resulting closed-loop system are determined by its closed-loop poles (ÅSTRÖM; HÄGGLUND, 2006). In this method, the controller's structure matches the complexity of the process model.

It consists of the following steps: (i) specify the poles that give the desired closed-loop behavior; (ii) compute the characteristic equation of the closed-loop system; (iii) compare, term by term, with the theoretical characteristic equation formed by a general controller and the process and (iv) compute the controller parameters that satisfy the constraints (ÅSTRÖM; WITTENMARK, 2013; CASTRUCCI; BITTAR; SALES, 2011). Based on the controller and process shown in Figure 40, the desired closed-loop poles can be specified as follows.

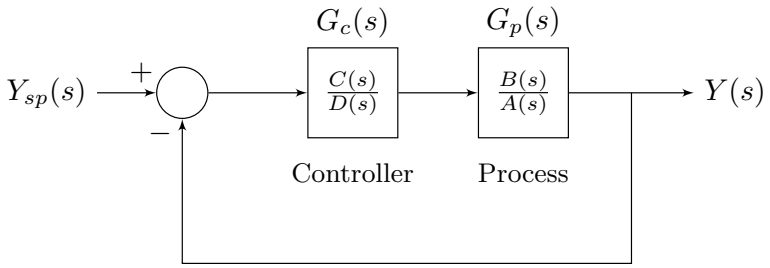


Figure 40 – Closed-loop system: controller and process for the APP method.

$$G_{\text{cl}}(s) = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} = \underbrace{\frac{B(s)C(s)}{A(s)D(s) + B(s)C(s)}}_{F(s)} \tag{A.2}$$

$$B(s) = b_0 + b_1 s + \cdots + b_n s^n \tag{A.3}$$

$$A(s) = a_0 + a_1 s + \cdots + a_n s^n,\ a_n \neq 0 \tag{A.5}$$

$$C(s) = c_0 + c_1 s + \cdots + c_m s^m \tag{A.4}$$

$$D(s) = d_0 + d_1 s + \cdots + d_m s^m,\ d_m \neq 0 \tag{A.6}$$

$$F(s) = f_0 + f_1 s + f_2 s^2 + \cdots + f_{(n+m)} s^{(n+m)} \tag{A.7}$$

The polynomials in Equation A.3-A.6 of the controller and the process must satisfy the following conditions: $\deg[B(s)] \leqslant \deg[A(s)]$ and $\deg[C(s)] \leqslant \deg[D(s)]$, with $n = \deg[A(s)]$ and $m = \deg[D(s)]$, then $\deg[F(s)] = n + m$. Substituting them in $F(s)$ and comparing the coefficients with similar powers, results in the system in Equation A.9, where the variables are the controller parameters.

$$\begin{aligned}&(a_0 + a_1 s + \cdots + a_n s^n)(d_0 + d_1 s + \cdots + d_m s^m) + \\ &\quad (b_0 + b_1 s + \cdots + b_n s^n)(c_0 + c_1 s + \cdots + c_m s^m) = f_0 + f_1 s + \cdots + f_{(n+m)} s^{(n+m)}\end{aligned} \tag{A.8}$$

$$\underbrace{\left[\begin{array}{cc|cc|c|cc|cc} a_0 & b_0 & 0 & 0 & & 0 & 0 & 0 & 0 \\ a_1 & b_1 & a_0 & b_0 & & 0 & 0 & 0 & 0 \\ \vdots & \vdots & a_1 & b_1 & & 0 & 0 & 0 & 0 \\ a_n & b_n & \vdots & \vdots & & 0 & 0 & 0 & 0 \\ 0 & 0 & a_n & b_n & \cdots & a_0 & b_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & & a_1 & b_1 & a_0 & b_0 \\ 0 & 0 & 0 & 0 & & \vdots & \vdots & a_1 & b_1 \\ 0 & 0 & 0 & 0 & & a_n & b_n & \vdots & \vdots \\ 0 & 0 & 0 & 0 & & 0 & 0 & a_n & b_n \end{array}\right]}_{(n+m+1)\times(2(m+1))} \underbrace{\left[\begin{array}{c} d_0 \\ c_0 \\ d_1 \\ c_1 \\ d_2 \\ c_2 \\ \vdots \\ d_m \\ c_m \end{array}\right]}_{2(m+1)} = \left[\begin{array}{c} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{(n+m)} \end{array}\right] \tag{A.9}$$

By design, the method places only the closed-loop poles, it does allow control over the system's closed-loop zeros. As a general controller tuning method, it applies to arbitrary controller structures, with PI and PID controllers resulting from first and second-order process model or by imposing the values of some of the controller's coefficients during the design phase.

A.6 Direct Synthesis

In the Direct Synthesis (DS) method, the controller is designed based on the process model and on the desired closed-loop dynamics (SEBORG *et al.*, 1989, p. 300). The procedure is illustrated in Figure 41 and Equation A.10. The controller $C_d(s)$ is analytically computed from the process model $P(s)$ and the desired closed-loop TF $G^d_{\text{cl}}(s)$, so that the desired closed-loop response is guaranteed (CHEN; SEBORG, 2002).

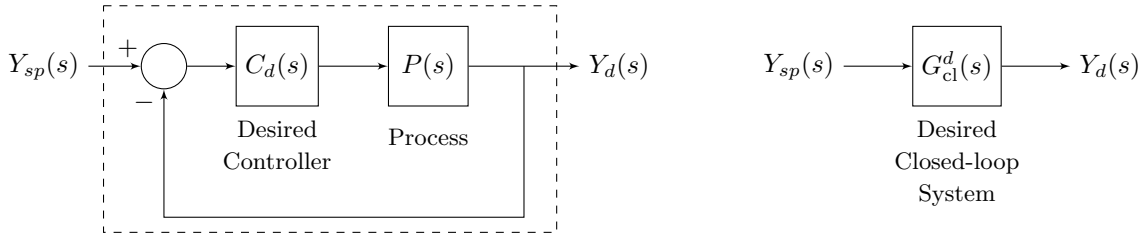


Figure 41 – Feedback system for the DS method.

$$G^d_{\text{cl}}(s)(s) = \frac{Y_d(s)}{Y_{sp}(s)} = \frac{C_d(s)P(s)}{1 + C_d(s)P(s)} \Longrightarrow C_d(s) = \frac{1}{P(s)} \frac{G^d_{\text{cl}}(s)}{1 - G^d_{\text{cl}}(s)} \tag{A.10}$$

The method cancels most of the process dynamics and includes the necessary dynamics to match the desired closed-loop response. Thus, unlike the APP method, it places both the zeros and the poles of the closed-loop system, respecting some constraints on the dynamics canceling:

1. $G^d_{\text{cl}}(s)$ must have, at least, the same relative degree of the process $P(s)$
2. $G^d_{\text{cl}}(s)$ must have, at least, the same time delay of the process $P(s)$
3. The non-minimum phase zeros of the process $P(s)$ must be zeros of $G^d_{\text{cl}}(s)$
4. The unstable poles of the process $P(s)$ must be zeros of $1 - G^d_{\text{cl}}(s)$
5. $G^d_{\text{cl}}(s)$ must satisfy the performance requirements for the transient and steady-state regime, which must not be too stringent

It is a general controller design method that for some common process models results in PI and PID controllers (SEBORG *et al.*, 1989).

A.7 Lambda Tuning

The λ-Tuning method, originally proposed by Dahlin (1968), is a particular case of pole placement in which the process is approximated by a FOTD model (ÅSTRÖM; HÄGGLUND, 2006). It is also a special case of the DS method (CHEN; SEBORG, 2002). According to Åström and Hägglund (1995), it is an analytical method suitable for processes with high dead-time. Its name comes from the desired closed-loop time constant λ, a design parameter that controls the response time (ÅSTRÖM; MURRAY, 2021). It assumes that the closed-loop behavior is given by a FOTD model as shown in Figure 42.

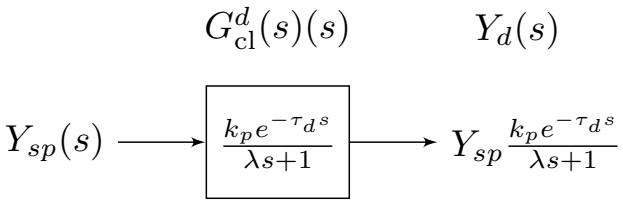


Figure 42 – Block diagram of the desired closed-loop system for the λ-Tuning method.

Due to its formulation, the poles and zeros of the process are canceled by the poles and zeros of the controller (ÅSTRÖM; HÄGGLUND, 1995). Furthermore, the method requires knowledge of the process model since the controller parameters are calculated analytically. An advantage of the method is that it is applicable to systems with non-minimum phase zeros and to systems with low damping (ÅSTRÖM; HÄGGLUND, 2006). The tuning rules for PI and PID controllers are given in Table 8.

A.8 Internal Model Control

The Internal Model Control (IMC) is a general controller tuning method that can be used for PID design (ÅSTRÖM; HÄGGLUND, 2006). It aims to obtain a controller with an appropriate structure and parameters from the process model and the desired requirements

Table 8 – λ-Tuning rules for PI and PID controllers for process modeled by a FOTD.

	PI	Series-PID	Parallel-PID
k_c	$\frac{1}{k_p}\frac{\tau}{\tau_d+\lambda}$	$\frac{1}{k_p}\frac{\tau}{\tau_d/2+\lambda}$	$\frac{1}{k_p}\frac{\tau_d/2+\tau}{\tau_d/2+\lambda}$
T_i	τ	τ	$\tau+\tau_d/2$
T_d	-	$\frac{\tau_d}{2}$	$\frac{\tau\tau_d}{\tau_d+2\tau}$
λ	$[\tau, 3\tau]$ or τ_d		

Source: Adapted from Segovia, Hägglund and Åström (2014a).

(RIVERA; MORARI; SKOGESTAD, 1986), with the closed-loop time constant as one of the design parameters (RIVERA; MORARI; SKOGESTAD, 1986).

In this method, the controller structure includes a low-pass filter G_f, in series with an estimate of the inverse of the process model $\hat{P}^{-1}(s)$, and in parallel the process model itself $\hat{P}(s)$ (ÅSTRÖM; HÄGGLUND, 1995), as shown in Figure 43. Hence the name internal model control. From this architecture, the controller as a whole is still connected in series with the process (ÅSTRÖM; HÄGGLUND, 2006).

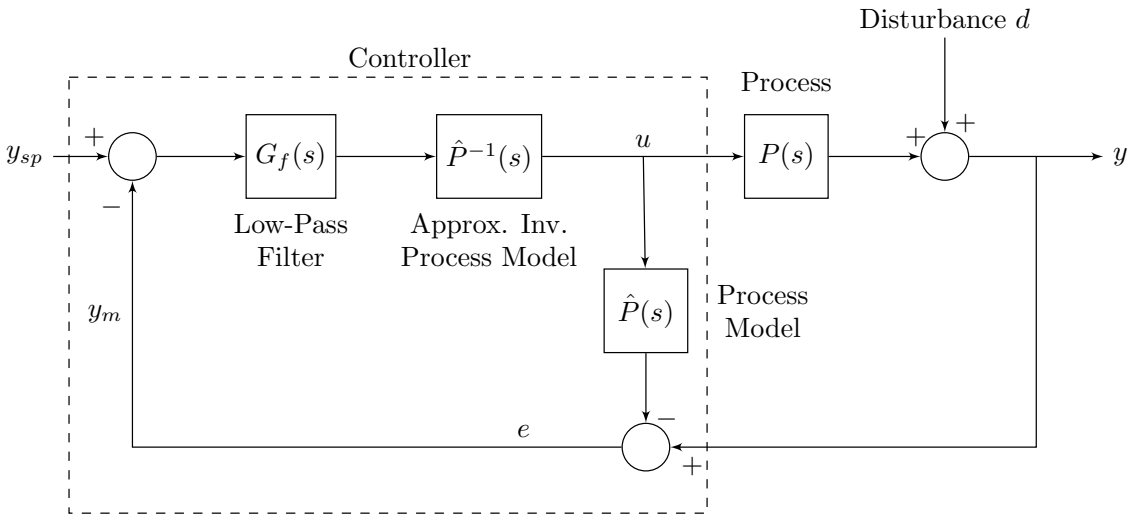


Figure 43 – Block diagram of a controlled system with the IMC tuning method.

Source: Adapted from Åström and Hägglund (2006).

The method considers the robustness in the controller design, with the trade-off between performance and robustness given by the filter time constant T_f (ÅSTRÖM; HÄGGLUND, 1995). The low-pass filter G_f is generally of first-order and reduces the system's sensitivity to modeling errors (ÅSTRÖM; HÄGGLUND, 2006).

Additionally, there are no restrictions on the type of process it can be applied to (RIVERA; MORARI; SKOGESTAD, 1986). A drawback of the method is the canceling of poles

and zeros of the process, unavoidable given the controller's internal structure (ÅSTRÖM; HÄGGLUND, 1995). Furthermore, it results in higher-order controllers for first-order process models, with PI and PID controller forms depending on the time-delay approximation adopted (ÅSTRÖM; HÄGGLUND, 2006).

For instance, considering the TFs in Equation A.11 and the controller given by the IMC method in Equation A.12.

$$P = \hat{P} = \frac{k_p e^{-\tau_d s}}{\tau s + 1}, \quad \hat{P}^{-1} = \frac{\tau s + 1}{k_p}, \quad G_f = \frac{1}{T_f s + 1}, \quad e^{-\tau_d s} \approx \begin{cases} 1 - \tau_d s & \text{(PI)} \\ \frac{1 - s\tau_d/2}{1 + s\tau_d/2} & \text{(PID)} \end{cases} \tag{A.11}$$

$$C(s) = \frac{G_f(s)\hat{P}^{-1}(s)}{1 - G_f(s)\hat{P}^{-1}(s)\hat{P}(s)} \tag{A.12}$$

If $e^{-\tau_d s}$ is approximated by a truncated taylor series expansion, the result is the PI controller in Equation A.13.

$$C_{\text{PI}}(s) = \frac{1 + \tau s}{k_p s(T_f + \tau_d)} = [k_p(T_f + \tau_d)]^{-1} \left(\frac{1}{s} + \tau\right) \tag{A.13}$$

And approximating it using a first-order padé approximation, results is the PID controller in Equation A.14.

$$C_{\text{PID}}(s) \approx \frac{(1 + s\tau_d/2)\,(1 + \tau s)}{k_p s(T_f + \tau_d)} \tag{A.14}$$

A.9 Skogestad Internal Model Control

The Skogestad Internal Model Control (SIMC) method is an improved version of the IMC that avoids the cancellation of poles from the process, making the desired closed-loop response time proportional to the process dead-time (SKOGESTAD, 2003). This method also uses the desired closed-loop time constant as the single design parameter (SKOGESTAD, 2003).

The SIMC provides a set of simple and effective rules for controller tuning, shown in Table 9, using FOTD and Second-Order Time Delay (SOTD) models, Equation A.15, that approximate the process (CASTRUCCI; BITTAR; SALES, 2011). It is similar to the pole placement method, but in this case, the closed-loop TF is fully specified in terms of its poles and zeros (ÅSTRÖM; HÄGGLUND, 2006).

Skogestad (2003) highlights that as it is an analytical method, it requires obtaining a mathematical description of the process. Either by process modeling or by experimental data, obtaining a FOTD or SOTD model for the process (SKOGESTAD, 2003).

$$P(s) = \frac{k_p w_n^2 e^{-\tau_d s}}{s^2 + 2\zeta w_n s + w_n^2} \tag{A.15}$$

where k_p, τ_d, w_n, and ζ are the steady-state gain, the time delay, the natural frequency, and the damping ratio, respectively.

Table 9 – SIMC for PI and PID controllers for process modeled by a FOTD.

	PI	PID
k_c	$\frac{1}{k_p}\frac{\tau+\tau_d/3}{\lambda+\tau_d}$	$\frac{1}{3k_p\tau_a}\frac{3\tau_a+\tau_d}{\lambda+\tau_d}$
T_i	$\min\{\tau+\tau_d/3, 4(\lambda+\tau_d)\}$	$\tau_a+\tau_b$
T_d	-	$\frac{\tau_b}{1+\tau_b/\tau_a}$
λ	τ_d	$\tau_d/2$
τ_a	$\min\{\tau, 4(\lambda+\tau_d)\}$	
τ_b	$\tau_d/3$	

Source: Adapted from Segovia, Hägglund and Åström (2014a).

A.10 M-constrained Integral Gain Optimization

The M-constrained Integral Gain Optimization (MIGO) is a robust loop shaping method that seeks to optimize the load disturbance rejection capacity while imposing constraints on the system's maximum sensitivity (ÅSTRÖM; PANAGOPOULOS; HÄGGLUND, 1998). The authors Åström and Hägglund (2006) consider it one of the greatest contributions to control theory in face of its capacity to guarantee robustness, by allowing its inclusion as a requirement in the controller design project.

The method treats the problem of finding the optimal controller parameters as an optimization problem (ÅSTRÖM; PANAGOPOULOS; HÄGGLUND, 1998). Formulated as the maximization of the integral gain $k_i = k_c/T_i$, or equivalently, the minimization of the IE $(\text{IE} = 1/k_i)$, subjected to robustness constraints to uncertainties in the process model, captured by the desired maximum sensitivity M_s (ÅSTRÖM; HÄGGLUND, 2006).

These constraints translate into constraints on the controller parameters (ÅSTRÖM; HÄGGLUND, 2006). The maximization of the integral gain is reduced to the solution of a set of non-linear algebraic equations (ÅSTRÖM; PANAGOPOULOS; HÄGGLUND, 1998). While the constraint on robustness, the maximum sensitivity allowed, is reduced to a circular region, around the critical point -1 in the complex s-plane, which should not be encircled by the Nyquist curve of the loop transfer function (ÅSTRÖM; PANAGOPOULOS; HÄGGLUND, 1998).

The method's main steps can be summarized as follows: (i) it starts with a fixed derivative gain k_d and the parameters and controller gains are calculated through the equations resulting from the restrictions; (ii) compute the sensitivity value for this set of solutions and test whether $M \geqslant M_s$ and (iii) increase the derivative gain k_d to its highest possible value until the robustness constraint is violated (ÅSTRÖM; HÄGGLUND, 2006). The solution of the numerical problem uses the Newton-Raphson method to search for solutions

and the Bisection method to obtain the best initial solutions (ÅSTRÖM; PANAGOPOULOS; HÄGGLUND, 1998).

The method requires knowledge of the process TF. It seeks to reject low-frequency disturbances, and for reference tracking purposes, it requires set-point weighting (ÅSTRÖM; HÄGGLUND, 2006). Furthermore, it applies to various types of systems: stable and integrating processes, with large and small dead-time, real and complex poles, and with positive and negative zeros (ÅSTRÖM; PANAGOPOULOS; HÄGGLUND, 1998). Finally, it requires a numerical solution and does not result in formulas or equations for controller parameters based on the process characteristics.

A.11 Approximated M-constrained Integral Gain Optimization

It is an extension of the MIGO method, with its acronym derived from *Approximated*-MIGO (ÅSTRÖM; HÄGGLUND, 2006). This method provides tuning rules for PI controllers, covered in Hägglund and Åström (2002) and Hägglund and Åström (2004), and for PID in Åström and Hägglund (2004). The tuning rules are based solely on the process characteristics obtained by the step or the frequency response methods (ÅSTRÖM; HÄGGLUND, 2006). Thus, it combines the ZN experiments with the robust loop shaping method MIGO (ÅSTRÖM; HÄGGLUND, 2004).

The AMIGO tuning rules were formulated by analyzing the correlation between a set of optimized controller parameters and the process response, which is characterized by its static gain k_p, dead-time τ_d, and the normalized time constant τ (ÅSTRÖM; HÄGGLUND, 2004). The method uses more information about the process than the ZN since requirements for load disturbance rejection and robustness to modeling errors are considered in the controller design (ÅSTRÖM; HÄGGLUND, 2006).

For a specified maximum sensitivity, one of the method's design parameters, simple and general tuning rules for PI controllers are available for the main industrial processes dynamics (HÄGGLUND; ÅSTRÖM, 2002). For PID controllers, simple and more general tuning rules are given only for more conservative tunings, which can result in lower performance (ÅSTRÖM; HÄGGLUND, 2004; ÅSTRÖM; HÄGGLUND, 2006). When compared with other methods like ZN, CC and λ-Tuning, the AMIGO method, at least for PI controllers, returns substantially better results, as one can see in the study of Hägglund and Åström (2002). The tuning rules for PI and PID for process modeled by FOTD are given in Table 10.

Other studies comparing PID tuning rules can be found in Foley, Julien and Copeland (2008) and Tan *et al.* (2006). In Foley, Julien and Copeland (2008), the IMC, SIMC, DS, and LEE tuning method (LEE *et al.*, 1998) are compared. In Tan *et al.* (2006), the ZN, CC, IMC,

Table 10 – AMIGO for PI and PID controllers for process modeled by a FOTD.

	PI	PID
k_c	$\frac{0.15}{k_p} + \frac{\tau}{k_p \tau_d}\left(0.35 - \frac{\tau_d \tau}{(\tau_d + \tau)^2}\right)$	$\frac{1}{k_p}\left(0.2 + 0.45\frac{\tau}{\tau_d}\right)$
T_i	$0.35\tau_d + \frac{13\tau_d\tau^2}{\tau^2 + 12\tau_d\tau + 7\tau_d^2}$	$\tau_d \frac{0.4\tau_d + 0.8\tau}{\tau_d + 0.1\tau}$
T_d	-	$\frac{0.5\tau_d\tau}{0.3\tau_d + \tau}$

Source: Adapted from Segovia, Hägglund and Åström (2014a).

Gain and Phase Margin (GPM) method (ÅSTRÖM; HÄGGLUND, 1984), IAE & Integral Time Absolute Error (ITAE) optimal tuning (HO *et al.*, 1996), and Integral Squared Error (ISE) & Integral Squared Time Error (ISTE) (ZHUANG; ATHERTON, 1993) are compared.