**UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC**
**CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT**
**PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA ELÉTRICA – PPGEEL**

**LUAN VINÍCIUS FIORIO**

# VIRTUAL REFERENCE FEEDBACK TUNING WITH ROBUSTNESS CONSTRAINTS: A SWARM INTELLIGENCE SOLUTION

**JOINVILLE**
**2022**

**LUAN VINÍCIUS FIORIO**

# VIRTUAL REFERENCE FEEDBACK TUNING WITH ROBUSTNESS CONSTRAINTS: A SWARM INTELLIGENCE SOLUTION

Master's thesis presented to the Graduate Program in Electrical Engineering of the Santa Catarina State University, as requirement for obtention of the title of Master in Electrical Engineering, concentration area of Electronic Systems.
Supervisor: Prof. Yales Rômulo de Novaes, Dr.
Co-supervisor: Chrystian Lenon Remes, Dr.

**JOINVILLE**
**2022**

Fiorio, Luan Vinícius
Virtual Reference Feedback Tuning with robustness constraints: a swarm intelligence solution / Luan Vinícius Fiorio. - Joinville, 2022.
101 p. : il. ; 30 cm.

Supervisor: Prof. Yales Rômulo de Novaes, Dr.
Co-supervisor: Chrystian Lenon Remes, Dr.
Master's thesis - Santa Catarina State University, Center of Technological Sciences, Graduate Program in Electrical Engineering, Joinville, 2022.

1. Data-driven control. 2. Robust control. 3. Swarm intelligence algorithms. 4. Virtual Reference Feedback Tuning. I. de Novaes, Dr, Prof. Yales Rômulo . II. Remes, Dr, Chrystian Lenon. III. Santa Catarina State University, Center of Technological Sciences, Graduate Program in Electrical Engineering. IV. Title.

**LUAN VINÍCIUS FIORIO**

**VIRTUAL REFERENCE FEEDBACK TUNING WITH ROBUSTNESS CONSTRAINTS: A SWARM INTELLIGENCE SOLUTION**

Master's thesis presented to the Graduate Program in Electrical Engineering of the Santa Catarina State University, as requirement for obtention of the title of Master in Electrical Engineering, concentration area of Electronic Systems.
Supervisor: Prof. Yales Rômulo de Novaes, Dr.
Co-supervisor: Chrystian Lenon Remes, Dr.

**EXAMINATION BOARD**

Prof. Yales Rômulo de Novaes, Doctor
Santa Catarina State University (UDESC) (Board's president / Supervisor)

Members:

Prof. Alexandre Sanfelici Bazanella, Doctor
Federal University of Rio Grande do Sul (UFRGS)

Prof. Antonio da Silva Silveira, Doctor
Federal University of Pará (UFPA)

Prof. Sérgio Vidal Garcia Oliveira, Doctor
Santa Catarina State University (UDESC)

Joinville, 14th February, 2022

*À minha mãe, Alessandra.*

## ACKNOWLEDGEMENTS

Firstly, I thank UDESC, FAPESC, and other related organizations, for their infrastructural and financial support. In such a turbulent time, I hope that science may advance much further in this country, for there is no lack of good researchers.

I thank all the professors that I have had in this master's program for their valuable lectures, comments, and guidance. I shall highlight: Alessandro Luiz Batschauer; Sérgio Vidal Garcia Oliveira; Tiago Jackson May Dezuo; and Yales Rômulo de Novaes.

I thank the Electrical Engineering Graduate Program's Secretary, Cíntia Tacla Gehring Fadel, for making my life always much easier when it comes to documents and forms. I also thank the UDESC Graduate Programs' Secretariat for their help when needed.

I thank my colleagues that worked with me or assisted me during the execution of every report, simulation, and prototype during the whole master's, in which I highlight: Daniel Steil Alves; Eduardo Falchetti Sovrani; Fabrício Trentini; Leandro Jun Kimura; Lucas Bona; Vinícius Guilherme Hoffmann. I also thank my work colleagues from my time in WEG Industries, César Orellana Lafuente and Mateus Abreu de Andrade, for their support, friendship, and jokes.

A special thanks to the content creators that filled my lonely research days during the pandemic with joy and laughs, who I must highlight Luide Matos from Luideverso and Petko from The Dark Den.

I thank the coffee and the beer.

I thank whoever shot an arrow in my knee so I could not be an adventurer like you, but a researcher.

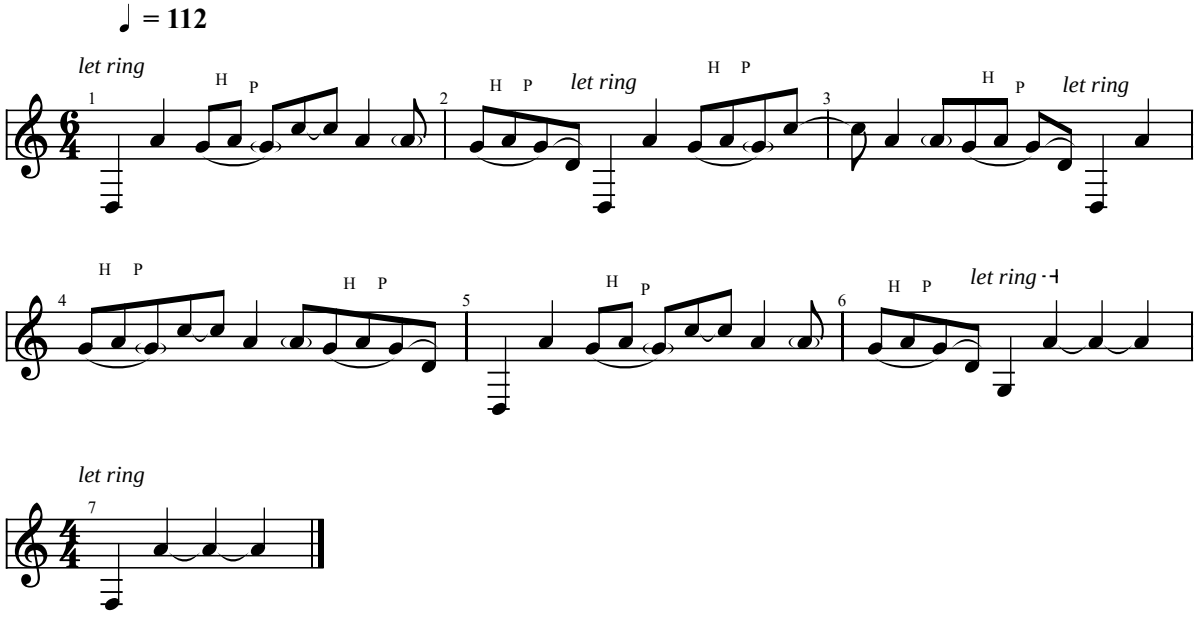
A stochastic thanks to Lucas Schulze, the colleague from another lab, that was virtually present, every day, in every discussion, every brainstorm, giving several insights, discussing anything that you can imagine, and making this home office life during the pandemic much better.

A huge thanks my closest personal friends (you know who you are) for being there. They are the reason that I smile every day.

I deeply thank my family for always being there for me with their kindness, love and support. Words are not enough to describe how deep this thanks goes.

Finally and most importantly, I thank my supervisors and masters, Chrystian Lenon Remes and Yales Rômulo de Novaes, for the guidance, teaching, friendship, and support. For their time and attention. For passing on what they have learned, strength, mastery, but weakness, folly, failure also. I am immensely grateful that our paths have crossed. Now, I shall carry on. As once said by a great master:

> "We are what they grow beyond. That is the true burden of all masters."
> Master Yoda, (JOHNSON; LUCAS, 2018).




Tool - Lateralus (JONES et al., 2002).

**ABSTRACT**

This work proposes an extension to the Virtual Reference Feedback Tuning (VRFT) method by adding an $\mathcal{H}_\infty$ robustness constraint to the optimization problem, which is solved by metaheuristic optimization. The estimation of the $\mathcal{H}_\infty$ norm of the sensitivity function is addressed in a data-driven fashion, based on the regularized estimation of the impulse response of a system. Among the available types of metaheuristics, four different swarm intelligence algorithms are chosen to be evaluated and compared. Two real-world inspired examples, inspired in the structure of dc-dc converters like Boost/Buck-Boost and SEPIC, are used to test and illustrate the proposed method.

**Keywords**: Data-driven control. Robust control. Swarm intelligence algorithms. Virtual Reference Feedback Tuning.

## RESUMO

Este trabalho propõe uma extensão ao Método da Referência Virtual adicionando uma restrição de robustez $\mathcal{H}_\infty$ ao problema de otimização, que é resolvido por otimização metaheurística. A estimativa baseada em dados da norma $\mathcal{H}_\infty$ da função de sensibilidade é feita a partir da estimativa da resposta ao impulso do sistema. Dentre as metaheurísticas disponíveis, quatro diferentes algoritmos de inteligência de enxame são escolhidos para serem aplicados ao problema proposto e comparados entre si. Dois exemplos inspirados na estrutura de conversores cc-cc como Boost/Buck-Boost e SEPIC são abordados para ilustrar o método proposto.



**Palavras-chave**: Controle baseado em dados. Controle robusto. Algoritmos de inteligência de enxame. Método da Referência Virtual.

## LIST OF FIGURES

## LIST OF TABLES

## LIST OF ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| ABC | Artificial Bee Colony |
| ACO | Ant Colony Optimization |
| CbT | Correlation-based Tuning |
| CCM | Continuous Conduction Mode |
| DC | Diagonal / Correlated |
| DD | Data-Driven |
| DD-LQR | Data-Driven Linear Quadratic Regulator |
| DI | Diagonal |
| DIT | Direct Iterative Tuning |
| FA | Firefly Algorithm |
| FIR | Finite Impulse Response |
| GM | Gain Margin |
| GWO | Grey Wolf Optimizer |
| IFT | Iterative Feedback Tuning |
| I-GWO | Improved Grey Wolf Optimizer |
| IR | Impulse Response |
| KH | Krill Herd |
| LQG | Linear Quadratic Gaussian |
| LQR | Linear Quadratic Regulator |
| LTR | Loop-Transfer-Recovery |
| MPE | Mean Percent Error |
| MRC | Model Reference Control |
| MSE | Mean Square Error |
| NFL | No Free Lunch |
| NMP | Non-Minimum Phase |
| OCI | Optimal Controller Identification |
| PI | Proportional-Integral |
| PID | Proportional-Integral-Derivative |
| PM | Phase Margin |
| PRBS | Pseudo-Random Binary Sequence |

| | |
|---|---|
| PSO | Particle Swarm Optimization |
| SISO | Single-Input Single-Output |
| SNR | Signal-to-Noise Ratio |
| TC | Tuned Correlated |
| VDFT | Virtual Disturbance Feedback Tuning |
| VRFT | Virtual Reference Feedback Tuning |
| WHK | Wiener-Hopf-Kalman |
| WOA | Whale Optimization Algorithm |

## LIST OF SYMBOLS

| | |
|---|---|
| $\mathcal{H}_2$ | System 2-norm |
| $\mathcal{H}_\infty$ | System $\infty$-norm |
| $\mathcal{H}_1$ | System 1-norm |
| $\gamma$ | Robustness index for $\mathcal{H}_\infty$ control |
| $G(z)$ | Transfer function of a discrete-time process |
| $y(k)$ | Output signal |
| $u(k)$ | Plant input signal / control output signal |
| $v(k)$ | Additive noise signal |
| $z$ | Forward time-shift operator |
| $k$ | Discrete-time sample |
| $GM$ | Gain margin |
| $PM$ | Phase margin |
| $f(\cdot)$ | Example cost function for optimization |
| $H(\cdot)$ | Penalty function |
| $B(\cdot)$ | Barrier function |
| $\Pi$ | Constraint set |
| $\delta$ | Threshold for defining convergence |
| $\Omega_{180}$ | Phase crossover frequency |
| $\Omega_c$ | Gain crossover frequency |
| $M_S$ | $\mathcal{H}_\infty$ norm of the sensitivity transfer function |
| $\dot{x}$ | Temporal derivative of the variable $x$ |
| $g(k)$ | Impulse response of $G(z)$ |
| $Y_N$ | Output regressor vector of length $N$ |
| $\Psi_N$ | Input regressor vector of length $N$ |
| $\psi(k)$ | Input signal regressors at instant $k$ |
| $\theta$ | Impulse response coefficients |
| $\varepsilon$ | Prediction error |
| $M$ | Number of terms of the impulse response |
| $X$ | Regularization matrix |
| $P_M$ | Covariance matrix |

| | |
|---|---|
| $\Lambda$ | Noise regressors |
| $N$ | Number of samples of a signal |
| $\mathcal{N}$ | Gaussian probability density function model |
| $\sigma$ | Variance |
| $I$ | Identity matrix |
| $\hat{\theta}_N^{apost}$ | *A posteriori* estimation of $\theta$ |
| $P_N^{apost}$ | *A posteriori* covariance matrix |
| $\hat{\theta}_N^{LS}$ | Non-regularized least squares |
| $P_{DC}$ | Diagonal/Correlated kernel |
| $P_{DI}$ | Diagonal kernel |
| $P_{TC}$ | Tuned Correlated kernel |
| $x(k)$ | Discrete-time signal $x$ at instant $k$ |
| $\mathcal{L}_p$ | Signal $p$-norm |
| $Y_M$ | Output signal represented in a matrix form |
| $U_M$ | Input signal represented in a matrix form of length $M$ |
| $G_M$ | Impulse response of $G(z)$ represented in a Toeplitz matrix form of length $M$ |
| $\bar{\sigma}$ | Largest singular value |
| $\lambda_{max}$ | Largest eigenvalue |
| $S(z)$ | Sensitivity transfer function |
| $T(z)$ | Complementary sensitivity transfer function |
| $r(k)$ | Reference signal |
| $\|\|\cdot\|\|$ | Norm operator |
| $C(z)$ | Controller transfer function |
| $e(k)$ | Error signal |
| $\mathcal{C}$ | Controller class |
| $\bar{C}(z)$ | Vector of rational functions which forms a linear parameterized controller |
| $k_p$ | Proportional gain |
| $k_i$ | Integral gain |
| $k_d$ | Derivative gain |
| $\rho$ | Controller parameters |
| $\mathcal{P}$ | Set of admissible values for $\rho$ |

| | |
|---|---|
| $J^{MR}$ | Model reference control cost function |
| $T_d(z)$ | Reference model |
| $C_d(z)$ | Ideal controller |
| $C(z,\rho)$ | Controller transfer function with parameters $\rho$ |
| $T(z,\rho)$ | Complementary sensitivity transfer function formed with the $C(z,\rho)$ controller |
| $\bar{r}(k)$ | Virtual reference signal |
| $\bar{e}(k)$ | Virtual error signal |
| $J^{VR}$ | Virtual Reference Feedback Tuning cost function |
| $\phi(k)$ | Signal composed of $\bar{C}(z)\bar{e}(k)$ |
| $L(z)$ | Transfer function of the filter for the mismatched case |
| $\Phi_r(e^{j\Omega})$ | Power spectra of $r(k)$ |
| $\Phi_u(e^{j\Omega})$ | Power spectra of $u(k)$ |
| $\phi_L(k)$ | Signal $\phi(k)$ filtered by $L(z)$ |
| $u_L(k)$ | Signal $u(k)$ filtered by $L(z)$ |
| $y_L(k)$ | Signal $y(k)$ filtered by $L(z)$ |
| $\eta$ | Parameters of the reference model |
| $F(z)$ | Vector of transfer functions of the reference model |
| $T_d(z,\eta)$ | Flexible reference model |
| $\phi_\eta(\rho,k)$ | VRFT with flexible criterion input regressor to estimate $\eta$ |
| $\tau_\eta$ | VRFT with flexible criterion output regressor to estimate $\eta$ |
| $\phi_\rho(\eta,k)$ | VRFT with flexible criterion input regressor to estimate $\rho$ |
| $\tau_\rho$ | VRFT with flexible criterion output regressor to estimate $\rho$ |
| $\hat{\rho}$ | Estimated controller parameters $\rho$ |
| $\hat{\eta}$ | Estimated reference model parameters $\eta$ |
| $\ell$ | Number of agents in a metaheuristic algorithm |
| $D$ | Dimension of the problem |
| $n$ | Iteration number in a metaheuristic algorithm |
| $\overrightarrow{X}_i$ | Position of an agent $i$ in a metaheuristic algorithm |
| $\overrightarrow{V}_i$ | Velocity of an agent $i$ in the PSO algorithm |
| $\overrightarrow{P}_i$ | Best found solution of a particle $i$ in PSO |

| | |
|---|---|
| $\overrightarrow{G}$ | Best global solution found by the swarm in PSO |
| $w_1$ | Inertia weight of the PSO algorithm |
| $w_2, w_3$ | Random variables of the PSO algorithm such that $w_2, w_3 \sim U(0,1)$ |
| $C_1$ | Cognitive learning factor of the PSO algorithm |
| $C_2$ | Social learning factor of the PSO algorithm |
| $l_b, u_b$ | Lower and upper search bounds, respectively |
| $U$ | Uniform probability density function model |
| $r_b$ | Random number such that $r_b \sim U(0,1)$ of the ABC algorithm |
| $a$ | Acceleration coefficient of the ABC algorithm |
| $r_a$ | Random number such that $r_a \sim U(-a,a)$ of the ABC algorithm |
| $\overrightarrow{X}_r$ | Random agent position for the ABC and the I-GWO algorithms |
| $P_i$ | Probability of an onlooker bee to choose a food source $\overrightarrow{X}_i$ in the ABC algorithm |
| $\overrightarrow{X}_\alpha$ | Position vector of the alpha wolf in the GWO algorithm |
| $\overrightarrow{X}_\beta$ | Position vector of the beta wolf in the GWO algorithm |
| $\overrightarrow{X}_\delta$ | Position vector of the delta wolf in the GWO algorithm |
| $\overrightarrow{X}_{\omega,k}$ | Position vector of a omega wolf $k$ in the GWO algorithm |
| $\overrightarrow{D}_p$ | Displacement vector in the GWO algorithm |
| $\overrightarrow{X}_{w\alpha}$ | Auxiliary position vectors of the $\alpha$ wolf in the GWO algorithm |
| $\overrightarrow{X}_{w\beta}$ | Auxiliary position vectors of the $\beta$ wolf in the GWO algorithm |
| $\overrightarrow{X}_{w\delta}$ | Auxiliary position vectors of the $\delta$ wolf in the GWO algorithm |
| $\overrightarrow{A}, \overrightarrow{C}$ | Coefficient vectors of the GWO algorithm |
| $r_1, r_2$ | Random vectors such that $r_1, r_2 \sim U(0,1)$ |
| $Pop$ | Population matrix of the grey wolves on the I-GWO algorithm |
| $r_j$ | Random variable such that $r_j \sim U(0,1)$ |
| $\overrightarrow{X}_{i,GWO}$ | Candidate position based on the GWO for the I-GWO algorithm |
| $\overrightarrow{R_i}$ | Euclidean distance between the current position and the candidate position in the I-GWO algorithm |
| $\overrightarrow{N}_i$ | Neighbors of $\overrightarrow{X}_i$ in the I-GWO algorithm |
| $\overrightarrow{X}_{i,DLH}$ | DLH-based candidate position in the I-GWO algorithm |
| $r_i$ | Random vector such that $r_i \sim U(0,1)$ |

| | |
|---|---|
| $S(z,\rho)$ | Sensitivity transfer function depending of $\rho$ |
| $\xi(k)$ | Input signal for estimating $\|\|S(z,\rho)\|\|_\infty$ With a single batch of data |
| $\zeta(k)$ | Output signal for estimating $\|\|S(z,\rho)\|\|_\infty$ with a single batch of data |
| $\hat{M}_S(\rho)$ | Estimated $\|\|S(z,\rho)\|\|_\infty$ norm |
| $M_{Sd}$ | Desired $\|\|S(z,\rho)\|\|_\infty$ norm |
| $J^{SI}$ | Swarm Intelligence cost function |
| $H(\rho)$ | Penalty term dependent of $\rho$ |
| $O$ | Search space within lower $l_b$ and upper $u_b$ bounds |
| $\rho_0$ | First step solution of the proposed method |
| $R$ | initial spawn radius |
| $G_1(z)$ | Second-order plant (Boost-based) transfer function from example 1 |
| $C_{IGWO}$ | I-GWO-designed controller used for illustration |
| $\rho_{IGWO}$ | I-GWO-designed controller parameters used for illustration |
| $G_2(z)$ | Fourth-order plant (SEPIC-based) transfer function from example 2 |

## CONTENTS

# 1 INTRODUCTION

The inherent complexity of several processes of different natures sometimes requires the designer to simplify the mathematical modeling of the plant (NISE, 2000), in order to save time or to in fact obtain a model, which may be too complex to be done without simplifications (CHAUDHURI; CHAKRABORTY; CHAUDHURI, 2012; XIE; KAMWA; CHUNG, 2021). The obtainment of a precise model, required to design a satisfactory controller for grid operation, is one example of a complex task related to the modeling phase (CHAUDHURI; CHAKRABORTY; CHAUDHURI, 2012). For some dc-dc converters, the control techniques assume the existence of a very accurate model (KAZIMIERCZUK, 2008; KOBAKU; PATWARDHAN; AGARWAL, 2017), which is a challenge to the designer, since power converters in general are nonlinear. Another situation that presents some difficulty to the designer is the obtention of low order controllers for more complex plants, which are vastly applied in industry (AGUIAR et al., 2018; THARANIDHARAN et al., 2022; TUDON-MARTINEZ et al., 2022; TAN et al., 2022). This difficulty can be originated from a poor modeling, since the process is complex, and/or from the limited performance of the chosen controller structure (KEEL; BHATTACHARYYA, 2008).

Most of the aforementioned problems can be suppressed with the use of data-driven control design techniques (REMES et al., 2021a; ZENELIS; WANG, 2022; HUANG et al., 2022), which overcome the need of a model as well as model-originated dilemas, such as the dilema on representativity and complexity, since important aspects of the system are contained in data and are weighted accordingly through the optimization criteria. Some data-driven approaches require several batches of data (i.e., several plant experiments) as Iterative Feedback Tuning (HJALMARSSON, 1998) and Iterative Correlation-based Tuning (KARIMI; MIŠKOVIĆ; BONVIN, 2004). On the other hand, methods as the Virtual Reference Feedback Tuning (VRFT) (CAMPI; LECCHINI; SAVARESI, 2002), the Virtual Disturbance Feedback Tuning (VDFT) (ECKHARD; CAMPESTRINI; BOEIRA, 2018), and the Optimal Controller Identification (OCI) (CAMPESTRINI et al., 2017) require only a single batch of data. Executing only a single experiment to design a controller in a data-driven fashion is a desirable feature, since it results in simpler experimentation, less memory requirements, and an overall less tedious process. Among those, the VRFT poses as a simpler algorithm that can be used to obtain a controller with a desired reference tracking dynamics. Thus, this work is based on the VRFT method (CAMPI; LECCHINI; SAVARESI, 2002; CAMPESTRINI et al., 2011; BAZANELLA; CAMPESTRINI; ECKHARD, 2012).

Considering low order controllers, robustness is a frequent topic of discussion (PÉREZ; LLOPIS, 2018; ALCÁNTARA; VILANOVA; PEDRET, 2013), since the simplified modelling may result in unsatisfactory controller performance, as well as it might not consider the process' uncertainties in its formulation. Some low order controller classes may not be able to achieve the required controller performance defined for the reference model. Despite the existence of design guidelines, it is not possible to know *a priori* if the chosen reference model is (exactly)

satisfactory and can really be achieved by the defined controller class. Such choices of reference model and/or controller class might lead to poor performance and poor robustness of the closed-loop system (BAZANELLA; CAMPESTRINI; ECKHARD, 2012). Recently, to suppress some of the aforementioned problems regarding robustness, the inclusion of a robustness criteria in the VRFT design in a systematic fashion has been done (CHILUKA et al., 2021), at the expense of: i) more experiments, since the proposed systemic design method regards a trial-and-error procedure in the VRFT design until the desired robustness value is achieved, which essentially removes one of the greatest advantages of the VRFT - being a one-shot method; and ii) iterative procedures like this usually require more background knowledge from the designer to choose a reference model and to set requirements of the closed-loop dynamic behavior.

Nevertheless, knowing that robustness can be measured by the $\mathcal{H}_\infty$ norm of the sensitivity transfer function of a closed-loop system (SKOGESTAD; POSTLETHWAITE, 2005), this work proposes the inclusion of an $\mathcal{H}_\infty$ norm constraint in the VRFT cost function maintaining its most attractive feature: the necessity of only a single batch of data. The constraint is added to the cost function in the form of a penalty (LUENBERGER; YE, 2015), spoiling the convex behavior of the VRFT cost function. Considering non-convex cost functions, classical optimization techniques often result in a solution that is not sufficiently close to the solution point (LUENBERGER; YE, 2015). For that reason, metaheuristic optimization is used in literature when a convex approximation cannot be obtained for a non-convex cost function (TALBI, 2009). Therefore, metaheuristics are applied to the considered problem in this work. The proposed method is composed of two main steps: i) the design, in a data-driven fashion, of a controller using the VRFT approach, if a previous controller is not already existent; and ii) considering the controller obtained at the previous step as initial solution and using the same batch of data, the application of a metaheuristic optimization algorithm to minimize the cost function considering an $\mathcal{H}_\infty$ robustness constraint.

Since metaheuristic algorithms may do well on average over a class of problems, but do worse on average over other class of problems, according to the No Free Lunch (NFL) theorems (WOLPERT; MACREADY, 1997), more than a single metaheuristic algorithm shall be considered. Among the different types of metaheuristics (MIRJALILI, 2019; ALATAS; CAN, 2015; WAHAB; NEFTI-MEZIANI; ATYABI, 2015), this work focuses on the application of swarm intelligence algorithms, since they contain the least number of hyperparameters to be chosen if compared to the other types (WAHAB; NEFTI-MEZIANI; ATYABI, 2015). Four swarm intelligence algorithms are considered: Particle Swarm Optimization (PSO) (KENNEDY; EBERHART, 1995); Artificial Bee Colony (ABC) (KARABOGA; BASTURK, 2007); Grey Wolf Optimizer (GWO) (MIRJALILI; MIRJALILI; LEWIS, 2014); and the most recent Improved Grey Wolf Optimizer (I-GWO) (NADIMI-SHAHRAKI; TAGHIAN; MIRJALILI, 2021).

In summary, the general objective of this thesis is the development of a robustness constrained-method regarding the VRFT cost function, which, considering the non-convex behavior caused by the inclusion of such constraint, will be minimized using swarm intelligence

algorithms. Within the main objective, some minor objectives may be stated:

- proposal of a method for estimating system norms depending only on the system's impulse response, which should simplify the already existent methods that, mostly, require more than a single experiment to estimate such norms;
- obtain the $\mathcal{H}_\infty$ norm of the sensitivity transfer function of a closed-loop system considering only a single-batch of data;
- application of the proposed method in two real world inspired plants, comparing the used swarm intelligence algorithms with each other in an statistical form, as well as comparing the results obtained with the proposed method to the results of the VRFT in terms of performance and robustness.

This thesis is structured as follows: Chapter 2 briefly reviews the available literature over the addressed topics; Chapter 3 proposes a method for estimating system norms in a data-driven fashion with a single batch of data, based on the estimation of impulse response coefficients in a regularized fashion; Chapter 4 details the Virtual Reference Feedback Tuning control design technique, including the case for non-minimum phase plants; Chapter 5 comments on the four chosen swarm intelligence algorithms, detailing their formulation; Chapter 6 explain the details of the proposed method of this thesis; Chapter 7 illustrate the proposed method through two real-world inspired examples; finally, Chapter 8 concludes this work and suggests future research.

# 2 LITERATURE REVIEW

This chapter addresses a review over the available literature of some relevant points to this thesis: robust control, data-driven control, system norms estimation, and optimization. In the next subsection, optimization-based control and robust control are addressed.

## 2.1 ROBUST AND OPTIMIZATION-BASED CONTROL

Some "classical" controller design techniques present a challenge if the control requirements are too rigorous, and/or the process has a too complex model, resulting in a more expensive procedure in terms of design time. In those cases, optimization techniques could be used to find the controller parameters (GREEN; LIMEBEER, 1994; APKARIAN; NOLL, 2019).

The Wiener-Hopf-Kalman (WHK) was one of the first control system optimization method, in the 1950s, when the United States and the Soviet Union were funding research into guidance and maneuvering of space vehicles (ZHOU; DOYLE, 1998). The WHK assumes that the plant has a precise linear model, possibly time varying, and that the noises and disturbances of the system have statistical properties that are known. One of its main applications was in resource management, minimizing the fuel needed to land a rocket by properly controlling the system. Once a cost function is settled in a quadratic form, the WHK procedure minimizes the problem resulting in a unique optimal controller, without requiring any further design intervention, i.e., without any fine adjusting or trial and error tasks. When the WHK was applied to industrial problems, a mismatch between its assumptions (e.g., precise model, statistical properties from disturbances and noise) and industrial applications led to the need of controller synthesis techniques that could deal with "non-precise" or *uncertain* models.

Robust control takes into account the uncertainty of a process considering mainly two types of problems: analysis, which determines if the controlled signals satisfy the desired robustness properties; and synthesis, that in fact designs the controller to satisfy such properties (ZHOU; DOYLE, 1998).

### 2.1.1 $\mathcal{H}_2$ optimal control

A system norm is a single number that contains information regarding gain, energy, robustness, among other possible physical interpretations, which can be used as a tool for system analysis and controller design (SKOGESTAD; POSTLETHWAITE, 2005). The $\mathcal{H}_2$ optimal control problem has the objective of finding a rational controller that stabilizes internally a plant and minimizes the $\mathcal{H}_2$ norm of the transfer function from reference to output (ZHOU; DOYLE, 1998). An example is the Linear Quadratic Regulator (LQR) problem, that has the objective of finding a control function for a dynamical system expressed in a state-space form, such that the states of the system are driven to a small neighbourhood of origin (ZHOU; DOYLE, 1998). The optimal solution of the LQR formulation guarantees at least 60° phase margin and

6 dB gain margin with the obtained controller. The limitation of the LQR is the choice of the weighted system matrices, which requires experience from the designer. In fact, plenty of works in the literature (DUPONT et al., 2013; LEUNG; TAM; LI, 1991; BHUSHAN; CHATTERJEE; SHANKAR, 2016) have addressed methods to find the weighting matrices of LQR.

Another example of $\mathcal{H}_2$ optimal control design procedure is the Linear Quadratic Gaussian (LQG) control (ÅSTRÖM, 1970; ATHANS, 1971), which assumes the presence of additive white Gaussian noise in the system, combining the LQR control law with a Kalman filter. For the case of the LQG, however, the guarantee of at least $60^\circ$ phase margin and 6 dB gain margin from the LQR is not maintained (DOYLE, 1978) and is dependent on the parameters (weighting matrices) choice. Later, a systematic approach, called Loop-Transfer-Recovery (LTR), to recover robustness properties for the LQG has been proposed for systems that are stable and minimum-phase (ATHANS, 1986). The finding of weighting matrices via optimization (REMES et al., 2021b) is also addressed in the literature regarding the LQG control design.

### 2.1.2 $\mathcal{H}_\infty$ optimal control

The optimal $\mathcal{H}_\infty$ control has the objective of finding all admissible controllers $C$ such that the $\mathcal{H}_\infty$ norm of a transfer function is minimized (ZHOU; DOYLE, 1998). Often, the $\mathcal{H}_\infty$ optimal controller presents greater difficulty to be found than $\mathcal{H}_2$ optimal controller (DOYLE et al., 1989). Therefore, in practice, it is much less expensive in terms of time and resources to design a *suboptimal controller*, which should be close to the optimal solution in a norm sense. The suboptimal $\mathcal{H}_\infty$ control problem is stated as: *given* $\gamma > 0$*, find all admissible controllers C, if they exist, such that* $||T||_\infty < \gamma$ (ZHOU; DOYLE, 1998).

The main forms of obtaining an $\mathcal{H}_\infty$ controller are: through the Youla-Kucera parametrization method (KUČERA, 2011), which leads to a high order controller; solving the Riccati differential equations, which require several assumptions that simplifies the problem and also require that the chosen $\gamma$ criterion does not have a too low value (GREEN; LIMEBEER, 1994); or using Linar Matrix Inequalities (LMIs) for solving the $\mathcal{H}_\infty$ optimal control problem (BOYD et al., 1994). The main downside of $\mathcal{H}_\infty$ controller design is that it is numerically and theoretically complicated, which may translate to more design time and more trials until the desired controller is obtained (SKOGESTAD; POSTLETHWAITE, 2005). While the "classic" $\mathcal{H}_\infty$ optimal control design guarantees robust stability, in order to optimize robust performance, the $\mathcal{H}_\infty$ loop-shaping method can be used (SKOGESTAD; POSTLETHWAITE, 2005), but deciding on a desired loop shape, depending on the problem, might be a difficult task to the designer.

Whilst for $\mathcal{H}_2$-based control design, uncertainties in the plant are not taken into account and a precise modeling is required to achieve high quality results in practice, techniques based on $\mathcal{H}_\infty$ optimal control may deal with model uncertainties. Another robust technique that has been extended to uncertain models is switched control, which is presented in the next subsection.

### 2.1.3 Switched control

A more recent control design technique based in optimization is the switched control, which models the system by modes, obtaining a control law that switches between each mode as needed, guaranteeing Lyapunov-based stability (TROFINO et al., 2009; COLANERI; GEROMEL; ASTOLFI, 2008; DEAECTO et al., 2010; EGIDIO; DEAECTO, 2019). Systems, such as power converters, can be modelled as affine[1] switched systems, allowing for the design of a switching rule that stabilizes the process and guarantee robustness (TROFINO et al., 2011). Uncertain parameters in the system can be considered in the modeling procedure by defining a polytope region for such parameter variation (BATTISTELLI; SELVI; TESI, 2017). Switched control has been also extended to include $\mathcal{H}_\infty$ optimization (TROFINO et al., 2012) for minimizing the effects of disturbances. The switching laws are designed in an offline fashion and are applied digitally to the process to be controlled. The main difficulty that can be noted within switched control is the modeling of the process as a switched system, which can become very complex depending on the plant's structure.

In some more complex cases, the obtention of a model is too expensive in terms of time and too exhausting to the designer. All of the aforementioned control design techniques rely on a mathematical model of the process to be controlled. Another design approach different from the aforementioned is data-driven control, which avoids the need of mathematical model of the plant, presenting itself as a viable solution to the existing gaps related to the modeling phase on the controller design. In the following, some available data-driven control techniques are presented.

## 2.2 DATA-DRIVEN CONTROL

The main characteristic of Data-Driven (DD) control design techniques is the fact that a mathematical model of the process to be controlled is not required. Some of the relevant DD methods are:

1. Iterative Feedback Tuning (IFT) (HJALMARSSON, 1998);
2. Direct Iterative Tuning (DIT) (KAMMER; BITMEAD; BARTLETT, 2000);
3. Virtual Reference Feedback Tuning (VRFT) (CAMPI; LECCHINI; SAVARESI, 2002);
4. Correlation-based Tuning (CbT) (KARIMI; MIŠKOVIĆ; BONVIN, 2004);
5. Optimal Controller Identification (OCI) (CAMPESTRINI et al., 2017);
6. Virtual Disturbance Feedback Tuning (VDFT) (ECKHARD; CAMPESTRINI; BOEIRA, 2018);
7. Data-driven LQR (DD-LQR) (SILVA et al., 2019),

[1] Nonlinear systems that are linear in the input.

The VRFT, OCI, VDFT, and DD-LQR, apart from the other techniques, have a distinguished feature: being one-shot methods, i.e., they require only a single batch of data (single experiment) to design the controller. The VRFT method is addressed in detail in Chapter 4.

The VDFT (ECKHARD; CAMPESTRINI; BOEIRA, 2018) uses input-output time-domain data from the process to design a controller based on a previously defined reference model for disturbances. For the design, a virtual disturbance signal is created, which justifies the name of the method. The VDFT results in a controller that, when operating in closed-loop with the process, should mimic the reference model behavior.

Correlation-based Tuning (KARIMI; MIŠKOVIĆ; BONVIN, 2004) decorrelates the output error between achieved and designed closed-loop system, iteratively tuning the controller parameters. The correlation equations are solved via stochastic approximation, converging to a unique solution. Since convergence may be too slow for practical applications, numerical algorithms are suggested to be used for solving the problem. The estimation of the gradient error, which is used for decorrelation, must be unbiased to guarantee the convergence of the algorithm. In order to obtain such estimation, a full-order model of the plant should be identified. In practice, an approximation of the gradient can be used, but without guarantees of convergence.

The OCI method (CAMPESTRINI et al., 2017) consists in solving a prediction error identification problem. The controller structure is defined in two parts, a fixed part and an identifiable part. The designer can consider different designs by the choices done over the fixed part. This method, besides dealing with controller classes both linear and nonlinear in the parameters, can provide an unbiased estimation of the controller parameters with less variance than other solutions, e.g., VRFT, at the cost of increased complexity in the optimization problem. On the other hand, the data-driven solution for the LQR problem - DD-LQR (SILVA et al., 2019) - obtains the state feedback gain by estimating a sequence of Markov parameters and an observability matrix from data, solving a quadratic problem, but it still relies on the good choice of the weighting matrices by the designer.

Although, such one-shot methods provide an optimal solution for a given performance criterion, robustness is not *explicitly* considered on them. Therefore, the next subsection presents some recent robust approaches for data-driven control.

### 2.2.1 Robust data-driven control

Some recent data-driven approaches address robustness criterion in the design of the controller. In the literature, a recent methodology (CHILUKA et al., 2021) proposed the systematization of the VRFT design considering a robustness constraint. The problem with the proposed method is that it resembles a trial-and-error solution, requiring several experiments in order to achieve the desired robustness value, which essentially eliminates the greatest advantage of the VRFT over other DD control design techniques: being one-shot.

A data-driven approach to robust control regarding $\mathcal{H}_2$, $\mathcal{H}_\infty$, and loop-shaping specifications has been recently addressed (KARIMI; KAMMER, 2017; NICOLETTI; MARTINO;

KARIMI, 2019), with convex optimization via LMIs. Such method, although, leads to higher-order controller solutions and it also relies on an initial solution that seem to have influence on the final one. Such approach is based on the frequency response of the system, lacking, in the literature, the development of a similar technique that uses only time-domain data.

Other robust data-driven solutions are done in an online fashion, which has the downside of requiring higher computational processing and having to measure the input-output data set at every iteration of the algorithm, e.g., the use of the modified algebraic Riccati equation with online data-driven learning (NA et al., 2021) or the application of a data-driven Model Predictive Control technique with robustness guarantees (BERBERICH et al., 2021).

The fact that the aforementioned one-shot data-driven approaches do not address robustness, and the robust data-driven control techniques require several batches of data and/or result in high order controllers, inspired the definition of the subject of this thesis, where a robustness constraint is included at the VRFT cost function in order to achieve a robust solution. The VRFT method is detailed in Chapter 4 and is chosen over OCI, VDFT, and DD-LQR since it has been more broadly applied to a several classes of problems, in order to attest the feasibility of the proposed method.

Once presented the overview of some relevant works existent in the literature of data-driven control and their robust extensions solutions, it is presented in the following section some fundamental aspects relevant to this thesis.

## 2.3 PRELIMINARIES: DESCRIPTION OF THE SYSTEM

The system considered in this thesis for the theoretical formulation is a discrete-time, causal, linear time-invariant, and Single-Input Single-Output (SISO) system $G(z)$, where $z$ is the forward discrete time-shift operator such that $zx(k) = x(k+1)$, whose output signal $y(k)$ can be described as

$$y(k) = G(z)u(k) + v(k), \tag{1}$$

where $u(k)$ is the input signal and $v(k)$ is the process noise - stochastic effects that are not represented by $G(z)$, i.e., that are not captured by the input-output relation between signals $u(k)$ and $y(k)$.

The closed-loop system taken into account in this thesis regards a controller $C(z)$ - which name might change depending on the design technique along the text - with the process $G(z)$, as shown in Figure 1, where $r(k)$ is the reference signal and $e(k)$ is the error signal. Any feedback gain or sensor is assumed to have insignificant dynamics, therefore it can be considered a constant gain, and can be included to the process model $G(z)$, making for a unitary feedback gain. The output of the closed-loop system is given as

$$y(k) = T(z)r(k) + S(z)v(k), \tag{2}$$

where the reference signal $r(k)$ is applied to the transfer function from $r(k)$ to the output $y(k)$, $T(z)$, with

$$T(z) = \frac{C(z)G(z)}{1+C(z)G(z)}, \tag{3}$$

and $S(z)$ is the sensitivity transfer function such that

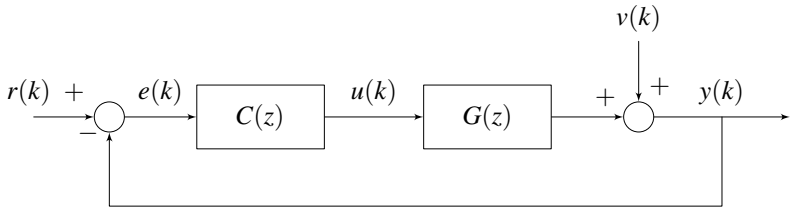
$$S(z) + T(z) = 1, \tag{4}$$

thus,

$$S(z) = \frac{1}{1+C(z)G(z)}. \tag{5}$$

Having the system defined, it is shown in the sequel the estimation of system norms.

Figure 1 – Block diagram of the considered closed-loop system structure for this thesis.




Source: the author.

## 2.4 ESTIMATION OF SYSTEM NORMS

The $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_\infty$ system norms are commonly used in robust and optimal control. Regarding the specific case of the $\mathcal{H}_\infty$ norm of the sensitivity transfer function ($M_S$), it is known that it is tied to the gain (GM) and phase margins (PM) of a system (SKOGESTAD; POSTLETHWAITE, 2005). Considering the open-loop transfer function $L(z) = G(z)C(z)$, the GM and PM can be, respectively, defined as:

$$GM = \frac{1}{|L(e^{j\Omega_{180}})|} \geq \frac{M_S}{M_S - 1}, \tag{6}$$

$$PM = \angle L(e^{j\Omega_c}) + 180^\circ \geq \frac{1}{M_S}, \tag{7}$$

where $\Omega = \omega T_s$ is the normalized frequency, $T_s$ is the sampling time, $\Omega_{180}$ is the phase crossover frequency, i.e., the frequency where the phase value is $180^\circ$, and $\Omega_c$ is the gain crossover frequency, i.e., the frequency where the magnitude value crosses 0 dB.

The very common data-driven approach is to estimate the norms with frequency response data (TACX; OOMEN, 2021), but to obtain such data set usually requires to execute several experiments. Iterative methods as Power Iterations and Weighted Thompson Sampling (MÜLLER; ROJAS, 2020) are also presented as a solution for the estimation of system norms at the expense of high computational cost.

Another way of estimating system norms relies in the estimation with the Toeplitz matrix of the system's Markov parameters (impulse response) (OOMEN et al., 2014). Recently, a similar approach to the estimation of the $\mathcal{H}_\infty$ norm with a different computational approach was used (SILVA; BAZANELLA; CAMPESTRINI, 2020). The method can be applied to both SISO or MIMO systems. This work inspires itself in such method and proposes the estimation of system norms directly via impulse response for SISO systems, since it can be estimated in a regularized fashion as presented in the literature (CHEN; OHLSSON; LJUNG, 2012) with the empirical Bayes method (CARLIN; LOUIS, 1997), as shown in Chapter 3, which counters the increased variance problem of the impulse response estimation (CHEN; OHLSSON; LJUNG, 2012).

With the estimation of system norms in a pure data-driven fashion, the inclusion of the $\mathcal{H}_\infty$ system norm constraint at the VRFT cost function can be done. Such inclusion results in a non-convex cost function, which can be minimized with optimization techniques. For that reason, the next section addresses optimization problems and algorithms.

## 2.5 OPTIMIZATION

Complex decision problems can be approached by focusing on a single objective designed to quantify performance and measure the quality of the decision, which can be minimized or maximized, depending on the formulation, subject to constraints (LUENBERGER; YE, 2015).

Optimization can be divided in two main parts: i) Linear programming - a problem that is characterized by linear functions of the unknowns. The objective is linear and the constraints are linear equalities or linear inequalities in the unknowns; and ii) Nonlinear programming - divided in unconstrained and constrained problems. The general mathematical programming problem can be written as (LUENBERGER; YE, 2015)

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & x \in \Pi, \end{aligned} \tag{8}$$

where $x$ is a n-dimensional vector of unknowns, $f$ is a real valued function of $x$, and $\Pi$ is a subset of n-dimensional space. The found solution does not ever reach the exact solution point, but converges towards it. The operation is usually terminated if the reached point of convergence is sufficiently close to the solution point or when a maximum number of iterations is reached. The convergence points can be *local* or *global*. Local solution points (local minimum) occur when the algorithm converges to a point that is not sufficiently close to the solution point and would take an exorbitant amount of time to reach the exact global solution. The global solution point (global minimum) is considered to be sufficiently close to the problem's solution.

Notice that in this work, the term *convergence* is considered to be when the difference of two subsequent evaluations of the cost function is lower than a certain value $\delta << 1$, e.g., $\delta = 1 \times 10^{-9}$.

Constrained problems can be approximated to unconstrained problems, commonly done via penalty or barrier methods (LUENBERGER; YE, 2015). Regarding penalty methods, the

constrained problem (8) can be changed by adding a *penalty* term to the main cost function, as

$$\underset{x}{\text{minimize}} \quad f(x) + cH(x), \tag{9}$$

where $c$ is a positive constant and $H$ is a function that satisfies: i) $H$ is continuous; ii) $H(x) \geq 0 \, \forall x$; and iii) $H(x) = 0$ if and only if $x \in \Pi$. Considering a large $c$, the minimum point of the problem (9) will be in a region where $H$ is small. Ideally, for $c \to \infty$ the solution point of the unconstrained problem with a penalty converges to a solution of the constrained problem.

Another form of adapting a constrained cost function to an unconstrained problem are the barrier methods (LUENBERGER; YE, 2015). A *barrier function* $B$ is defined in the interior of $\Pi$ in (8), such that: i) $B$ is continuous; ii) $B(x) \geq 0$; and iii) $B(x) \to 0$ as $x$ approaches the boundary of $\Pi$. The barrier function is included into the cost function as

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) + \frac{1}{c}B(x) \\ \text{subject to} \quad & x \in \text{interior of } B. \end{aligned} \tag{10}$$

Notice that, in (10), a constraint still exists but it is simpler to be implemented than the "fully" constrained problem (8).

Since some of the techniques applied in this work rely on linear or bilinear optimization problems, the next subsection presents the least squares criterion, which is an optimization technique that usually fits such type of problems.

### 2.5.1 Least squares criterion

For linearized problems, such as the estimation of the impulse response of a system around a specific operation point, an optimization criterion with analytic solution, named least squares, can be used to identify unknown parameters according to input-output data. The standard solution is based on a linear regression that considers the output of the process as

$$y(k) = \psi(k)\theta + v(k). \tag{11}$$

Considering $Y_N = [y(M+1)\ y(M+2)\ ...\ y(N)]'$ the $N$ output regressors, $\Psi_N = [\psi(M+1)\ \psi(M+2)\ ...\ \psi(N)]$, with $\psi(k) = [u(k-1)\ u(k-2)\ ...\ u(k-M)]'$, the $N$ input regressors, $\theta$ the $M$ parameters to be identified, and $\Lambda_N$ the N noise regressors $\Lambda_N = [v(k+1)\ v(k+2)\ ...\ v(N)]'$, with Gaussian distribution and variance $\sigma^2$, i.e., $v(k) \sim \mathcal{N}(0, \sigma^2)$, expression (11) can be rewritten in a more compact form by stacking $N - M + 1$ samples,

$$Y_N = \Psi_N'\theta + \Lambda_N. \tag{12}$$

Expression (12) represents the stacked signals specifically for the problem of impulse response estimation with $M$ coefficients (CHEN; OHLSSON; LJUNG, 2012) which is of interest for this thesis. To generalize the least squares criterion, $Y_N$, $\Phi_N$, and $\Lambda_N$ can be defined with $M = 0$. Also, note that the term *regressors* comes from the fact that $Y_N$, $\Psi_N$, and $\Lambda_N$ are *linear regressions*,

alluding to the fact that they are calculated or described "backwards" to the vectors that are used to compose them (LJUNG, 1999).

For notational simplicity, $\Lambda_N$ is considered to be null in the least squares formulation (LJUNG, 1999). The prediction error is given as

$$\varepsilon = Y_N - \Psi_N'\theta. \tag{13}$$

The objective of the regression is defined to reduce the prediction error in a 2-norm sense, problem which can be written in the form of a cost function:

$$\underset{\theta}{\text{minimize}} \quad ||Y_N - \Psi_N'\theta||_2^2. \tag{14}$$

The minimum argument of (14) represents the least squares solution of the problem, i.e., the solution that minimizes the squared prediction error. From (12), considering $\Lambda_N$ null, it is given that

$$Y_N = \Psi_N'\theta. \tag{15}$$

Since $\Psi_N'$ is not a square matrix, it cannot be inverted. Although, multiplying both sides of (15) by $\Psi_N$:

$$\Psi_N Y_N = \Psi_N \Psi_N'\theta, \tag{16}$$

now where $\Psi_N\Psi_N'$ is a square matrix and can be inverted. Isolating $\theta$ in (16), the solution is obtained as

$$\hat{\theta} = [\hat{g}_1\ \hat{g}_1\ ...\ \hat{g}_M]' = (\Psi_N\Psi_N')^{-1}\Psi_N Y_N, \tag{17}$$

where $(\Psi_N\Psi_N')^{-1}\Psi_N$ is known as the pseudo-inverse of $\Psi_N$.

The least squares can be used in some of the approached problems of this thesis, the inclusion of an $\mathcal{H}_\infty$ constraint at the VRFT optimization problem (as Chapter 6 details) results in a non-convex problem, with an unknown structure since the $\mathcal{H}_\infty$ norm is identified during optimization. Metaheuristic optimization is known to perform well in such situations (TALBI, 2009), thus, the following subsection presents a brief review over metaheuristics.

### 2.5.2 Metaheuristics

While classical methods try to find an exact solution to a problem, metaheuristics (or *heuristics*, in general) find an approximate solution, which is very useful to more complex optimization problems or unknown structures (black boxes) (TALBI, 2009). The approximate solution found by a metaheuristic algorithm should be as close as possible to the global minimum (LUKE, 2013), but there is no guarantee of convergence to the optimal global minimum (TALBI, 2009). Metaheuristics are seen as computational intelligence algorithms for optimization, and therefore, are considered to be a branch of artificial intelligence (DU; SWAMY, 2016).

In general, the most commonly used metaheuristics can be classified into three main classes: evolutionary algorithms; physics-based algorithms; and swarm intelligence algorithms

(MIRJALILI; MIRJALILI; LEWIS, 2014). Since literature over the topic is very wide and not necessarily have a convention over the classification of metaheuristic algorithms, several ways of defining and, thus, classifying an algorithm are presented below, according to (TALBI, 2009):

- **nature inspired versus nonnature inspired:** evolution, artificial immune systems, colonies of bees, ants, social sciences over animal species, and physics are took as inspiration for many of the nature inspired algorithms, whilst some algorithms (e.g., Greedy Randomized Adaptive Random Procedures - GRARP (FEO; RESENDE, 1995)) are nonnature based.

- **memory usage versus memoryless methods:** whilst some algorithms do not extract information dynamically during the search, others use a memory that contains information extracted during optimization, e.g., Tabu search (GLOVER; LAGUNA, 1998) and GRARP;

- **deterministic versus stochastic:** some metaheuristics solve problems by making deterministic decisions (e.g., Tabu search), whilst others contain some random rules (e.g., Simulated Annealing (KIRKPATRICK; GELATT; VECCHI, 1983), evolutionary algorithms, swarm intelligence algorithms);

- **population-based search versus single-solution based search:** single-solution algorithms (e.g., Simulated Annealing) manipulate and transform a single solution over the whole optimization procedure and are *exploitation oriented* - the search is intensified in local regions, i.e., the solution is focused on a single part of the search space at a time. Population-based algorithms, on the other hand, evolve a whole population of solutions and are *exploration oriented*, allowing for better diversification in the search space, i.e., there are several search agents (which represent solutions) spread throughout the search space;

- **iterative versus greedy:** in iterative algorithms, an initial complete solution is used as initialization and it is modified over the optimization procedure. For greedy algorithms, the initial solution is null.

Focusing on swarm intelligence algorithms, they are represented by a class of optimization algorithms of collective intelligence that mimics social behavior of animals, i.e., collective behavior of decentralized agents, resulting in local interactions of individual components as well as interactions with their environment and with the whole swarm (DU; SWAMY, 2016). This work addresses the use of such algorithms over other metaheuristics for the proposed problem for the following reasons (MIRJALILI; MIRJALILI; LEWIS, 2014): i) they preserve information about the search space over iterations, while evolutionary algorithms discard the information at each generation; ii) more often this type of algorithm uses memory to record the best found solution so far; iii) usually there are fewer parameters to adjust when compared to

other metaheuristics; iv) there are less operators when compared to evolutionary approaches (crossover, mutation, elitism, etc); and v) simpler implementation if compared to evolutionary algorithms and some physics based metaheuristics. Some of the most relevant swarm intelligence algorithms from 1995 (PSO) to 2021 (I-GWO) are listed below:

- Particle Swarm Optimization (PSO) (KENNEDY; EBERHART, 1995);
- Ant Colony Optimization (ACO) (DORIGO; BIRATTARI; STUTZLE, 2006);
- Artificial Bee Colony (ABC) (KARABOGA; BASTURK, 2007);
- Firefly Algorithm (FA) (YANG, 2010);
- Krill Herd (KH) (GANDOMI; ALAVI, 2012);
- Grey Wolf Optimizer (GWO) (MIRJALILI; MIRJALILI; LEWIS, 2014);
- Whale Optimization Algorithm (WOA) (MIRJALILI; LEWIS, 2016);
- Improved Grey Wolf Optimizer (I-GWO) (NADIMI-SHAHRAKI; TAGHIAN; MIRJALILI, 2021).

This work uses four of the listed swarm intelligence algorithms in order to minimize the proposed problem and compare the algorithms to each other, since the No Free Lunch (NFL) theorems (WOLPERT; MACREADY, 1997) states that if an algorithm does well on average for a class of problems, it might not for other class of problems. Therefore more than one metaheuristic optimization algorithm should be tested. Two of the algorithms were chosen because they are the most famous and commonly used in metaheuristic optimization, being PSO and ABC. The other two were chosen as more "modern" algorithms, the first being the GWO, which has presented very interesting results in the literature and has the least number of hyperparameters[2] over all cited algorithms. The I-GWO was the last chosen algorithm and represents an improvement over the GWO regarding the lack of population diversity, imbalance between exploitation and exploration, and premature convergence. Although there are plenty of improved versions of the GWO, the I-GWO still maintains the least number of parameters characteristic of the GWO, as well as it is a result of analyzing all other improved algorithms and composing a new one (NADIMI-SHAHRAKI; TAGHIAN; MIRJALILI, 2021). The chosen algorithms are described in details in Chapter 5.

## 2.6 FINAL CONSIDERATIONS

Control design based in optimization is usually required to reduce design time for more complex plants or to reach more rigorous control requirements. Historically, one of the first

[2] A parameter whose value is used to control the learning/optimization process and is usually chosen by the user.

optimization control design to be used was the Wiener-Hopf-Kalman for space craft applications. When this technique was applied in industry, some robustness problems appeared due to the poor modeling of the systems, requiring the development of robust techniques. The $\mathcal{H}_2$ optimal control is generally treated through the LQR and the LQG methods, and its primary focus is performance. On the other hand, the $\mathcal{H}_\infty$ optimal control is a class of design methods to obtain a controller that satisfies a $\gamma$ restriction of $\mathcal{H}_\infty$ norm, which leads to higher robustness, with the downside of more complex formulations resulting in little direct influence of the user over the design as well as still maintaining the need of a mathematical model to be obtained. Switching control is another robust approach based in optimization that requires the modeling of the system, but in a switched perspective, which can become very complex depending on the process' structure.

Data-driven control design methods do not require a model. Some methods like the Iterative Feedback Tuning and the Correlation-based Tuning require several experiments for obtaining a final solution, while the VRFT, OCI, VDFT, and DD-LQR are one-shot techniques that minimizes the $\mathcal{H}_2$ norm of defined optimization criterion. An attempt to include robustness criterion on the VRFT design (CHILUKA et al., 2021) resulted in a method that relies on several experiments, losing one of the most attractive VRFT feature. A data-driven approach to robust control via LMIs has been recently proposed (KARIMI; KAMMER, 2017; NICOLETTI; MARTINO; KARIMI, 2019), but results in a higher-order controller and the final solution is influenced by the initial considered solution. The inclusion of a robustness constraint in a one-shot technique, holding such characteristic, seems to be a gap in the literature. Such addition would result in a non-convex optimization problem, which can be solved via metaheuristic optimization, that are suitable for this type of problem.

The techniques for estimation of system norms available in the literature rely on methods that require several experiments or that need to estimate the whole state-space of an unknown system. A technique relying solely on the identification of the impulse response of the system, avoiding state-space identification, could simplify the problem and the formulation.

# 3 ESTIMATION OF SYSTEM NORMS

Throughout this chapter, a form to estimate system norms via impulse response is developed in order to allow the inclusion of a robustness constraint in the Virtual Reference Feedback Tuning (VRFT) cost function, since the $\mathcal{H}_\infty$ norm of the sensitivity transfer function is a direct measure of robustness (SKOGESTAD; POSTLETHWAITE, 2005). Firstly, the non-parametric estimation of impulse response coefficients is addressed. In the sequence, an approximated estimation of system norms, considering the estimated system response, is presented.

## 3.1 NON-PARAMETRIC ESTIMATION OF IMPULSE RESPONSE COEFFICIENTS

The output of the system $G(z)$ can be described by

$$y(k) = G(z)u(k) + v(k), \tag{18}$$

where $u(k)$ is the input signal and $v(k)$ is an additive noise term. The system $G(z)$ can be expressed considering its Impulse Response (IR) $g(i)$, $i \in \mathbb{N}$, as

$$G(z) = \sum_{i=1}^{\infty} g(i)z^{-i}. \tag{19}$$

Expression (19) can be truncated to a finite number, resulting in a Finite Impulse Response (FIR) model, where $g(i)$ can be estimated using least squares (CHEN; OHLSSON; LJUNG, 2012). Such truncation represents an approximation of the system $G(z)$,

$$G(z) \approx \sum_{i=1}^{M} g(i)z^{-i}, \tag{20}$$

which is sufficiently representative if the number of terms $M$ is enough to characterize the impulse response. Considering a stable system, $\lim_{k\to\infty} g(k) = 0$, which means that the IR tends to zero when its number of terms tends to infinite.

When using the standard least squares algorithm, as presented in Subsection 2.5.1, for the estimation of impulse response, the estimated coefficients $\hat{\theta}$ have their variance linearly increased with the order $M$, i.e., the higher the $M$ value, the poorer will be the estimation at the final terms of the IR because of the increasing variance (CHEN; OHLSSON; LJUNG, 2012). In order to estimate the impulse response more accurately, the issue of increased variance should be countered by including a compensation term in the least squares problem (14), that seeks for a solution that balances both the error and the norm of the parameter vector. In practice, this tends to reduce the influence of unimportant terms, by weighting the solution in the cost function, i.e., regularizing the solution, which can be represented in the form of an optimization problem similar to (14), given by

$$\min_{\theta} \sum_{k=M+1}^{N} (y(k) - \psi'(k)\theta)^2 + \theta' X \theta, \tag{21}$$

where $\theta' X \theta$ is the regularization term and $X$ is the regularization matrix. In (21), the solution to be found should be one that minimizes the Mean Square Error (MSE) of the estimates (first term) as well as the two-norm of the parameter vector (regularization term), which should result in a sparser solution than for the standard least squares solution (17). A sparser solution is one that has less non-zero values (or more zero values) than the original solution.

The solution for (21) can be obtained through a Bayesian perspective (CHEN; OHLSSON; LJUNG, 2012), which means to consider the parameter to be estimated a random variable, seeking some prior distribution of it given the observations. In order to do so, a Gaussian distribution for the parameter vector is considered as follows:

$$\theta \sim \mathcal{N}(0, P_M), \tag{22}$$

with zero mean and covariance matrix $P_M$, where $\theta$ are the parameters to be estimated. Recalling (12) (Chapter 2), it is also known that the output regressors are defined as

$$Y_N = \Psi_N' \theta + \Lambda_N, \tag{23}$$

where $\Psi_N$ are input regressors, $\Psi_N = [\psi(k+1)\ \ \psi(k+2)\ \ ...\ \ \psi(N)]$, and $\Lambda_N$ are the noise regressors composed of $\Lambda_N = [v(k+1)\ v(k+1)\ ...\ v(N)]$ where $v(k) \sim \mathcal{N}(0, \sigma^2)$. In order to regularize the estimation, since the variable to be estimated is considered a random variable in the Bayesian approach, $Y_N$ and $\theta$ are taken as jointly Gaussian random variables, i.e., variables in the same sample space with a joint characteristic function and a Gaussian joint density. By considering that, a Gaussian probability density function of the two variables can be defined (CHEN; OHLSSON; LJUNG, 2012) as

$$\begin{bmatrix} \theta \\ Y_N \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} P_M & P_M \Psi_N \\ \Psi_N' & \Psi_N' P_M \Psi_N + \sigma^2 I_{N-n} \end{bmatrix} \right). \tag{24}$$

Nevertheless, an *a posteriori* distribution of $\theta$ given $Y_N$ can be obtained from (24) as (CHEN; OHLSSON; LJUNG, 2012)

$$\theta | Y_N \sim \mathcal{N}(\hat{\theta}_N^{apost}, P_N^{apost}), \tag{25}$$

which represents the probability density function of $\theta$ given all the information contained in $Y_N$, where $\hat{\theta}_N^{apost}$ is given as (CHEN; OHLSSON; LJUNG, 2012)

$$\hat{\theta}_N^{apost} = ((\sigma^2 (\Psi_N \Psi_N')^{-1})^{-1} + P_M^{-1})^{-1} (\sigma^2 (\Psi_N \Psi_N')^{-1})^{-1} \hat{\theta}_N^{LS}, \tag{26}$$

and the covariance matrix $P_N^{apost}$ is (CHEN; OHLSSON; LJUNG, 2012)

$$P_N^{apost} = ((\sigma^2 (\Psi_N \Psi_N')^{-1})^{-1} + P_M^{-1})^{-1}, \tag{27}$$

in which the superscript $apost$ indicates *a posteriori*, the 'hat' marker stands for estimated values, and $\hat{\theta}_N^{LS}$ is the non-regularized case solution, obtained via (17). When $X = \sigma^2 P_M^{-1}$ in (21), $\hat{\theta}_N^{apost}$ is equal to the regularized estimate (CHEN; OHLSSON; LJUNG, 2012).

The covariance matrix $P_M$ can be defined according to some prior covariance matrices with predefined structure, also known as kernels. Three kernels are chosen to be used - Diagonal/Correlated (DC), Diagonal (DI), and Tuned Correlated (TC) - since they are used in the base literature for this work (CHEN; OHLSSON; LJUNG, 2012), given respectively by:

$$P_M = \begin{cases} P_{DC}(k,j) = c\iota^{|k-j|}\lambda^{(k+j)/2}; \\ P_{DI}(k,j) = \begin{cases} c\lambda^k, & if\ k=j; \\ 0, & otherwise; \end{cases} \\ P_{TC}(k,j) = c\ min(\lambda^j, \lambda^k), \end{cases} \tag{28}$$

which hyperparameters are defined in a vector form, for the DC kernel $\alpha = [c\ \lambda\ \iota\ \sigma]$, and for DI and TC kernels $\alpha = [c\ \lambda\ \sigma]$. From (24), isolating $Y_N$,

$$Y_N \sim \mathcal{N}(0, \sigma^2 I_{N-M} + \Psi_N' P_M(\alpha)\Psi_N). \tag{29}$$

The estimation of $\alpha$ in expression (29) can be done through the maximum likelihood approach (CARLIN; LOUIS, 1997), as

$$\hat{\alpha} = \min_{\alpha}\ Y_N'\Sigma(\alpha)^{-1}Y_N + \log\det\Sigma(\alpha), \tag{30}$$

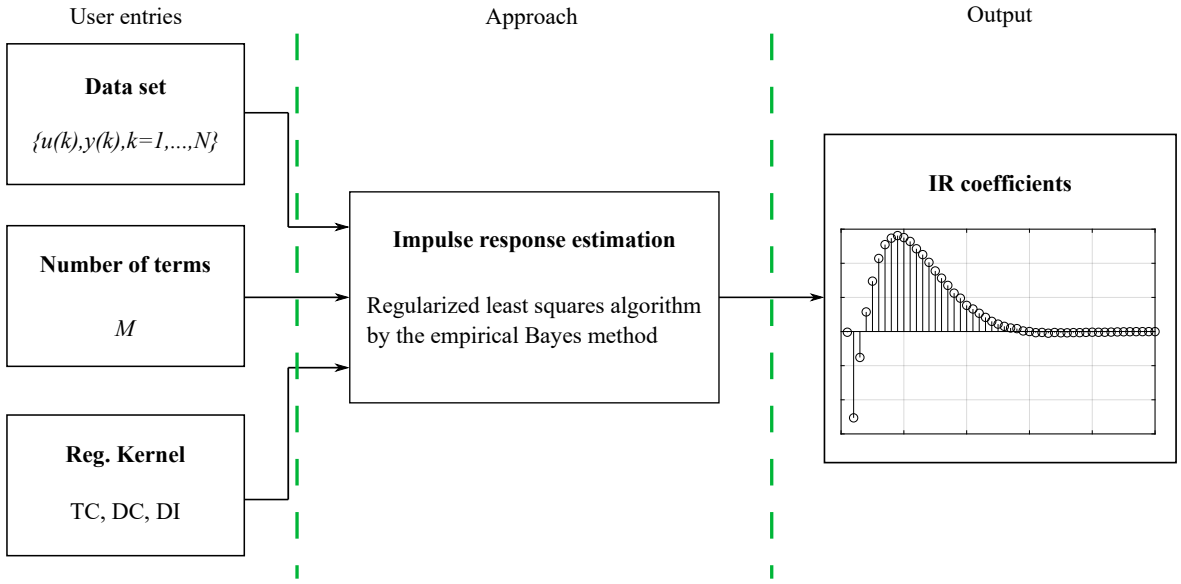
where

$$\Sigma(\alpha) = \sigma^2 I_{N-M} + \Psi_N P_M(\alpha)\Psi_N'. \tag{31}$$

This method for estimating the hyperparameters in the prior distribution is defined by (CARLIN; LOUIS, 1997) as the *empirical Bayes* method, which has already been used, e.g., for control design (BOEIRA; ECKHARD, 2018). The estimation in (30) can be done by using available classical optimization methods, available in some libraries, e.g., for Python - the Scipy Optimize library (COMMUNITY, 2020). The impulse response estimation method through empirical Bayes method is summarized in a flowchart diagram in Figure 2 . During the development of this thesis, the IR estimation was implemented as a package in Python, generating a publication (FIORIO; REMES; NOVAES, 2021), whose results are detailed in Appendix A.

The next section presents the estimation of system norms relying solely on the identified impulse response of the system.

Figure 2 – Flowchart diagram summarizing the impulse response estimation method through empirical Bayes method.




Source: the author.

## 3.2 ESTIMATION OF SYSTEM NORMS VIA IMPULSE RESPONSE

The signal norms $\mathcal{L}_p$ and $\mathcal{L}_\infty$ are defined as (SKOGESTAD; POSTLETHWAITE, 2005, A.5):

$$\mathcal{L}_p : ||x(k)||_p = \left( \sum_{k=0}^{\infty} |x(k)|^p \right)^{1/p}, \tag{32a}$$

$$\mathcal{L}_\infty : ||x(k)||_\infty = \max |x(k)|, \tag{32b}$$

where $x(k)$ is any time domain signal and $p \geq 1$, typically 1 or 2. The $\mathcal{L}_p$ norm, with $p = 2$, can be used to represent energy in a signal, for example, whilst the $\mathcal{L}_\infty$ is the peak value of the signal in time. The notation $\mathcal{L}$ is due to the fact that, for the continuous case of (32a), the integrand should be *Lebesgue*-integrable for the integral to exist (TOIVONEN, 2010).

On the other hand, for system norms, consider the system $G(z)$, with its output $y(k)$ described by the convolution between an input signal $u(k)$ and the impulse response coefficients $g(k)$,

$$G : y(k) = g(k) * u(k) = \sum_{n=0}^{\infty} g(k-n)u(M). \tag{33}$$

The main used norms are $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_\infty$, and are defined as (SKOGESTAD; POSTLETHWAITE, 2005, A.5):

$$\mathcal{H}_1 : ||G||_1 = \sum_{k=0}^{\infty} |g(k)| = \max_{u(k)\neq 0} \frac{||g(k) * u(k)||_\infty}{||u(k)||_\infty}, \tag{34a}$$

$$\mathcal{H}_2 : ||G||_2 = \sqrt{\sum_{k=0}^{\infty} |g(k)|^2} = ||g(k)||_2, \tag{34b}$$

$$\mathcal{H}_\infty : ||G||_\infty = \max_{u(k)\neq 0} \frac{||g(k) * u(k)||_2}{||u(k)||_2}. \tag{34c}$$

In this case, the notation used is $\mathcal{H}$ instead of $\mathcal{L}$ and refers to the fact that the function spaces have finite $\mathcal{L}_p$ norms on the imaginary axis and are bounded and analytic functions in the right-half plane, i.e., no poles outside the unit circle for the discrete-time case, being called *Hardy* spaces (TOIVONEN, 2010).

Expressions (32) and (34) shall be modified to allow the use of limited data (i.e., a practical case), which can be done by considering that the system $G$ is stable, and thus $\lim_{k\to\infty} g(k) = 0$. Therefore, the convolution in (33) can be truncated to $M$ terms, assuming that the IR terms with an order greater than $M$ are negligible (sufficiently close to zero), resulting in

$$G : y(k) = \sum_{n=0}^{\infty} g(k-n)u(M) \approx \underbrace{\sum_{n=0}^{M} g(k-n)u(M)}_{|g(M+1)|<\varepsilon, \text{ with } \varepsilon\to 0^+} . \tag{35}$$

For the norms $\mathcal{H}_1$ and $\mathcal{H}_2$, the truncated convolution (35) can be considered, approximating the expressions (34a) and (34b), respectively, to

$$||G||_1 \approx \sum_{k=0}^{M} |g(k)|, \tag{36}$$

$$||G||_2 \approx \sqrt{\sum_{k=0}^{M} |g(k)|^2}, \tag{37}$$

which allows for the obtention of the norms $\mathcal{H}_1$ and $\mathcal{H}_2$ with a limited amount of data (an IR $g(k)$ with only $M$ terms).

On the other hand, the $\mathcal{H}_\infty$ norm stands for the maximum gain relation considering all sets of possible input signals in a system (SKOGESTAD; POSTLETHWAITE, 2005). Therefore, such norm cannot be practically obtained using expression (34c), since it is required to consider all possible input signals in its evaluation. As an alternative strategy, a matrix relation to $g(k)$ can be obtained in order to allow the use of induced norm properties. Expression (35) can be expanded to $M$ terms as

$$\begin{cases} y(0) = g(0)u(0) \\ y(1) = g(1)u(0) + g(0)u(1) \\ \vdots \\ y(M) = g(M)u(0) + \cdots + g(0)u(M), \end{cases} \tag{38}$$

which can be represented in a matrix form,

$$\underbrace{\begin{bmatrix} y(0) \\ y(1) \\ \cdots \\ y(M) \end{bmatrix}}_{Y_M} = \underbrace{\begin{bmatrix} g(0) & 0 & \cdots & 0 \\ g(1) & g(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g(M) & g(M-1) & \cdots & g(0) \end{bmatrix}}_{G_M} \underbrace{\begin{bmatrix} u(0) \\ u(1) \\ \cdots \\ u(M) \end{bmatrix}}_{U_M}, \tag{39}$$

where the multiplication between $G_M$ and the input vector $U_M$ results in the output vector $Y_M$, where matrix $G_M$ represents the system's IR in the form of a Toeplitz matrix. If the order $M$ is sufficiently high, it can be said that matrix $G_M$ sufficiently characterizes the impulse response $g(k)$ and, therefore, system $G$.

An interesting property of matrices is the induced norm, which have a close relationship to signal amplification in systems (SKOGESTAD; POSTLETHWAITE, 2005). From the representation presented in (39), it is clear that

$$Y_M = G_M U_M. \tag{40}$$

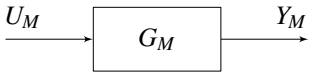
The "amplification" or "gain" of matrix $G_M$ is, then, given by the ratio $||Y_M||/||U_M||$. Therefore, the maximum gain considering all possible inputs, which is of particular interest, can be described as

$$||G_M||_{ip} = \max_{U_M \neq 0} \frac{||G_M U_M||_p}{||U_M||_p}, \tag{41}$$

where the subscript $i$ stands for induced. The p-norm of a matrix $A$, with elements $a_{ij}$ according to row $i$ and column $j$, can be calculated as follows (SKOGESTAD; POSTLETHWAITE, 2005, A.5):

$$||A||_p = \left( \sum_{i,j} |a_{ij}|^p \right)^{1/p}. \tag{42}$$

Figure 3 – Block diagram representation of (39).




Source: adapted from (SKOGESTAD; POSTLETHWAITE, 2005).

Since $G_M$ characterizes the system $G$, the induced norm can be used as an equivalent expression for (34c), since both represent the maximum gain considering all sets of input signals. Thus, relating (34c) to (41) with $p = 2$, results in

$$||G||_\infty = \max_{u(k) \neq 0} \frac{||g(k) * u(k)||_2}{||u(k)||_2} \approx \max_{U_M \neq 0} \frac{||G_M U_M||_2}{||U_M||_2} = ||G_M||_{i2}. \tag{43}$$

The induced-2 norm of a matrix, on the other hand, is also equivalent to the largest singular value of a matrix, which is obtained as

$$||G_M||_{i2} = \bar{\sigma}(G_M) = \sqrt{\lambda_{max}(G_M' G_M)}, \tag{44}$$

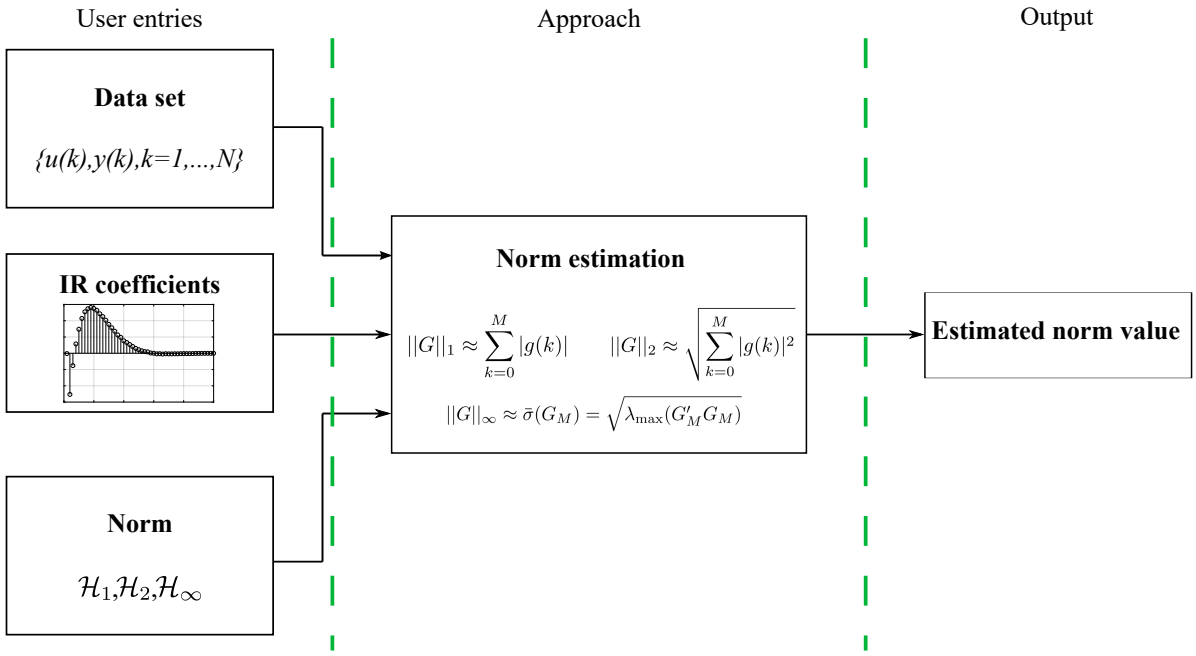
where $\bar{\sigma}$ stands for largest singular value and $\lambda_{max}$ is the largest eigenvalue of $G_M$. Finally, with (43) and (44), the $\mathcal{H}_\infty$ norm of $G$ can be approximated by

$$||G||_\infty \approx \bar{\sigma}(G_M) = \sqrt{\lambda_{max}(G'_M G_M)}, \tag{45}$$

which does not depend on a particular set of input signals, allowing for the $\mathcal{H}_\infty$ norm to be obtained directly from matrix $G_M$.

From the reasoning aforementioned, it is observed that the system norms can be calculated once the IR coefficients are estimated. Such coefficients, in turn, can be estimated with limited data in a regularized form with the empirical Bayes method, as presented in Section 3.1. In summary, the system norms can be obtained in a data-driven fashion as illustrated in Figure 4.

Figure 4 – Flowchart diagram summarizing the proposed norm estimation method based on impulse response.




Source: the author.

### 3.2.1 Estimating the norms of the sensitivity transfer function

In order to estimate the $\mathcal{H}_\infty$ norm of the sensitivity transfer function $S(z)$, since the proposed system norm estimation method is based on the estimation of the impulse response of the system, firstly, an expression for the input and output signals of $S(z)$ is needed to estimate its IR.

From the closed-loop system considered in this thesis, detailed in Subsection 2.3, consider an experiment such that

$$T(z) = \frac{C(z)G(z)}{1 + C(z)G(z)}, \tag{46}$$

where $S(z)$ is the sensitivity transfer function and relates to $T(z)$ as

$$S(z) + T(z) = 1, \tag{47}$$

the system's output can be described by

$$y(k) = T(z)r(k) + S(z)v(k). \tag{48}$$

The output signal can be rewritten considering the relation (47), then, as

$$y(k) = [1 - S(z)]r(k) + S(z)v(k) = r(k) + S(z)[v(k) - r(k)]. \tag{49}$$

Consequently,

$$y(k) - r(k) = S(z)[v(k) - r(k)], \tag{50}$$

resulting in

$$r(k) - y(k) = C^{-1}(z)u(k) = S(z)[r(k) - v(k)]. \tag{51}$$

Since the noise signal $v(k)$ does not appear in $C^{-1}(z)u(k)$ and it is usually too small to be measured separately, it is not considered in the data set used for estimation. Therefore, with the input-output set of data $\{r(k), C^{-1}(z)u(k), k = 1...N\}$, the estimation of the IR of $S(z)$ can be done as commented in Section 3.1 and, thus, the $||S(z)||_\infty$ can be estimated according to the method proposed in Section 3.2. The $\mathcal{H}_1$ and $\mathcal{H}_2$ norms of $S(z)$ can be obtained via expressions (34a) and (34b), respectively. The $\mathcal{H}_\infty$ norm of $S(z)$, on the other hand, can be estimated by the approximation $||S(z)||_\infty \approx ||S_M||_{i2}$, with $M$ sufficiently large, by forming a Toeplitz matrix $S_M$ as suggested in (39).

## 3.3 CASE STUDY

In order to evaluate the proposed method for estimating system norms via impulse response, a case study with the objective of obtaining the $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_\infty$ norms of five different systems is proposed. The norms are obtained using a Pseudo-Random Binary Sequence (PRBS) as excitation signal for the reference, since it is a persistent signal of high order (LJUNG, 1999), with $N = 2000$ samples. The IR is estimated by the method exposed in Section 3.1, with length $M = 100$ set arbitrarily, in a regularized fashion using the Tuned-Correlated kernel - as well as for all the examples provided in this case study - since it has less computational cost than the DC kernel, and is still precise enough to provide sufficiently accurate results (CHEN; OHLSSON; LJUNG, 2012). For each system in simulation, additive white Gaussian noise $v(k)$ with zero mean and a Signal-to-Noise ratio (SNR) of 10 dB was included at the process output and fed back to the system as shown in Figure 1.

The considered systems are presented in Table 1 and have structures that are commonly found in engineering problems (NISE, 2000). Tables 2, 3, and 4 show, respectively, the estimated $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_\infty$ norms, with its comparison to the real value (obtained by model), altogether

Table 1 – System's transfer functions $G(z)$ and controllers $C(z)$ used as examples.

| System | $G(z)$ | $C(z)$ |
|---|---|---|
| 1 | $\frac{0.5}{(z-0.9)}$ | $\frac{0.3797(z-0.9)}{(z-1)}$ |
| 2 | $\frac{-0.1(z-0.5)}{(z-0.9)(z-0.8)}$ | $\frac{-1.1600(z-0.9719)}{(z-1)}$ |
| 3 | $\frac{-0.05(z-0.6)}{(z^2-1.8z+0.82)}$ | $\frac{-3.7144(z-0.9351)(z-0.4210)}{z(z-1)}$ |
| 4 | $\frac{-0.05(z-1.4)}{(z-0.9)(z-0.8)}$ | $\frac{4.7942(z-0.9)(z-0.8)}{z(z-1)}$ |
| 5 | $\frac{3.605(z-0.55)(z^2-1.62z+0.6586)}{(z^2-1.84z+0.8564)(z^2-1.26z+0.4069)}$ | $\frac{0.0519(z-0.8977)}{(z-1)}$ |

Source: the author.

Table 2 – Each system's $\mathcal{H}_1$ norm of $S(z)$ calculated by model (Real), estimated via data (Data), and its percent error.

| System | Real | Data | Error (%) |
|---|---|---|---|
| 1 | 2.0000 | 2.0076 | 0.3825 |
| 2 | 2.0544 | 1.9466 | 5.2485 |
| 3 | 2.1656 | 2.1496 | 0.7358 |
| 4 | 2.4794 | 2.4778 | 0.0649 |
| 5 | 2.0307 | 1.9970 | 1.6578 |

Source: the author.

Table 3 – Each system's $\mathcal{H}_2$ norm of $S(z)$ calculated by model (Real), estimated via data (Data), and its percent error.

| System | Real | Data | Error (%) |
|---|---|---|---|
| 1 | 1.0511 | 1.0520 | 0.0784 |
| 2 | 1.0441 | 1.0428 | 0.1293 |
| 3 | 1.0542 | 1.0509 | 0.3166 |
| 4 | 1.0811 | 1.0846 | 0.3175 |
| 5 | 1.0523 | 1.0543 | 0.1889 |

Source: the author.

Table 4 – Each system's $\mathcal{H}_\infty$ norm of $S(z)$ calculated by model (Real), estimated via data (Data), and its percent error.

| System | Real | Data | Error (%) |
|---|---|---|---|
| 1 | 1.1049 | 1.1210 | 1.4635 |
| 2 | 1.1619 | 1.1521 | 0.8408 |
| 3 | 1.1272 | 1.1331 | 0.5186 |
| 4 | 1.5348 | 1.5375 | 0.1767 |
| 5 | 1.1006 | 1.1224 | 1.9781 |

Source: the author.

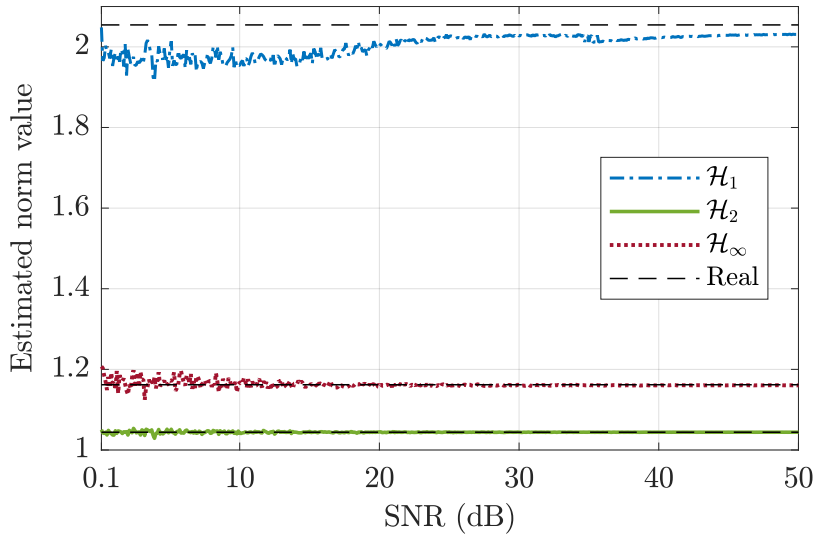
with the absolute value of the percent error between both values. The low percent errors found suggest good performance for estimating system norms with the proposed method.

In order to verify the influence of the noise amplitude to the norm estimation, the $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_\infty$ norms of the sensitivity transfer function are estimated for System 2 (Table 1), since it the second-order structure with minimum-phase zero is very common in control engineering

problems (NISE, 2000), with the SNR value varying from 0.1 to 50 dB, with a step of 0.1 dB. The result is presented in Figure 5. The reference signal considered for estimation is a PRBS, with the same characteristics as aforementioned.

To evaluate the quality of the estimation, the Mean Percent Error (MPE), in respect to all estimations in Figure 5, between the estimated and the real norm is taken into account, obtained as the average of the percent errors for each SNR value. For the estimation of the $\mathcal{H}_1$ norm, the obtained MPE is of 2.4287 %. For the $\mathcal{H}_2$ norm, an MPE of 0.0903 % is obtained. At last, the $\mathcal{H}_\infty$ norm estimation resulted in 0.3729 % of MPE. Visually, Figure 5 shows a less precise estimation for lower values of SNR, as expected, since it represents higher noise amplitude in relation to the signal amplitude. The $\mathcal{H}_1$ estimation is less precise than $\mathcal{H}_2$ and $\mathcal{H}_\infty$, which may be a result of the influence that each IR term has in expression (36), since every term is summed to the norm value.

Figure 5 – Estimated $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms of $S(z)$, for System 2, with its real values in dashed lines, for a wide SNR range.

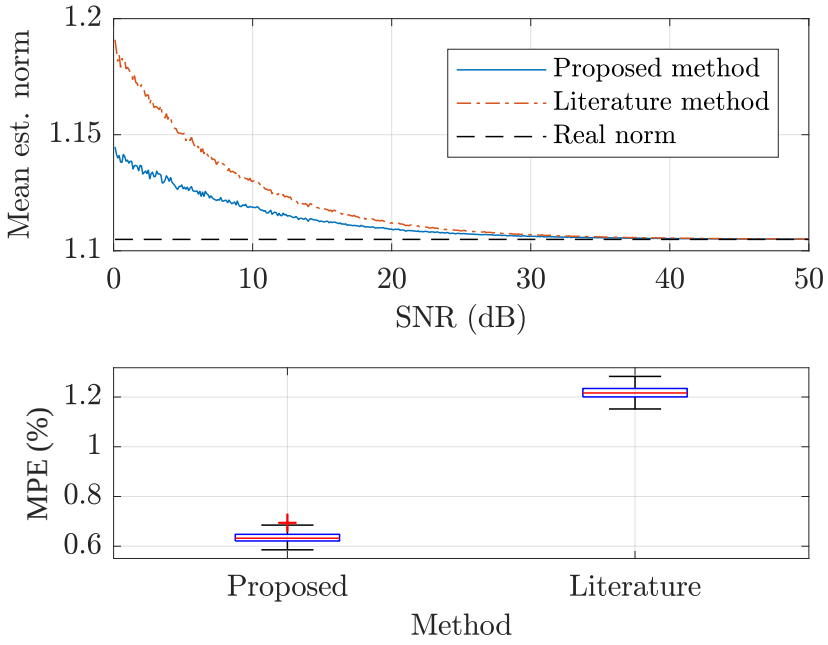



Source: the author.

The proposed norm estimation procedure is compared to a state-of-the-art approach from the literature (SILVA; BAZANELLA; CAMPESTRINI, 2020), which is inspired in state-space subspace system identification theory, for estimating the $\mathcal{H}_\infty$ norm of the sensitivity function of System 1, system chosen for its simplicity, since 100 runs are executed for each of the 500 considered values of SNR, demanding some substantial time to be executed. The number of estimates for the IR is chosen arbitrarily as $M = 100$, and the variation of SNR is considered as aforementioned. At each SNR value, a Monte Carlo experiment with 100 runs was executed, changing the noise realization at each run. Figure 6 shows the mean estimated norm for each SNR value with the proposed and the literature method, as well as a boxplot with the SNR statistics. It can be seen that the proposed method achieves a value closer to the real one, which is more perceptible for lower SNR values. The MPE for the proposed method was of 0.6351 %, whilst 1.2187 % for the literature method, which represents a reduction of 47.8871 % by the use

of the proposed approach. Such reduction is possibly influenced by the use of regularization in the IR estimation, which is not considered for the Markov parameter's estimation in the literature.

Figure 6 – Mean value of 100 Monte Carlo runs for the proposed and a literature method (SILVA; BAZANELLA; CAMPESTRINI, 2020), for estimating $||S(z)||_\infty$ of System 1, as well as its box plot representation.




Source: the author.

## 3.4 FINAL CONSIDERATIONS

In this chapter, the regularized estimation of impulse response was addressed and a method for estimating system norms based on the impulse response using only a single batch of data was proposed. The results show low values of percent error for all tested cases. In comparison to a state-of-the-art technique from the literature for estimating the $\mathcal{H}_\infty$ norm of a system (SILVA; BAZANELLA; CAMPESTRINI, 2020), the proposed method showed a considerable reduction of the mean percent error of the estimation for a system, which was observed in a Monte Carlo experiment throughout a wide range of SNR values. Some results of this chapter are published in the literature (FIORIO; REMES; NOVAES, 2021).

# 4 DATA-DRIVEN CONTROLLER DESIGN

In this chapter, data-driven controller design is addressed through the Virtual Reference Feedback Tuning (VRFT) method (CAMPI; LECCHINI; SAVARESI, 2002; CAMPESTRINI et al., 2011; BAZANELLA; CAMPESTRINI; ECKHARD, 2012), which is used by the method proposed in this thesis, and bases itself in the Model Reference Control (MRC), also explained here.

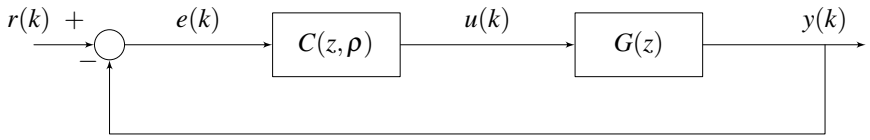
The closed-loop block diagram considered in this chapter, for reference tracking, is presented in Figure 7, where $C(z,\rho)$ is the controller to be designed, $r(k)$ is the reference signal, and $e(k)$ is the error signal. The noise signal $v(k)$ is neglected in the following expressions and from Figure 7 for simplicity. The output signal can be expressed in terms of the closed-loop transfer function from the reference signal $r(k)$ to the output signal $y(k)$ as

$$y(k) = T(z,\rho)r(k), \tag{52}$$

where

$$T(z,\rho) = \frac{C(z,\rho)G(z)}{1+C(z,\rho)G(z)}. \tag{53}$$

Figure 7 – Block diagram of the closed-loop system for reference tracking.




Source: the author.

The closed-loop control law, from Figure 7, is given as

$$u(k) = C(z,\rho)(r(k) - y(k)). \tag{54}$$

The controller is linearly parameterized as

$$C(z,\rho) = \rho'\bar{C}(z), \tag{55}$$

in which $\rho \in \mathbb{R}^p$ represent the parameters of the controller, e.g., for PID controller $\rho = [k_p\ k_i\ k_d]'$, where $k_p$, $k_i$, and $k_d$ are, respectively, the proportional, integral, and derivative controller gains. $\bar{C}(z)$ is a vector of transfer functions, e.g., in the case of the PID controller:

$$\bar{C}(z) = \begin{bmatrix} 1 \\ \frac{z}{z-1} \\ \frac{z-1}{z} \end{bmatrix} \tag{56}$$

such that

$$C(z,k_p,k_i,k_d) = k_p + k_i\frac{z}{z-1} + k_d\frac{z-1}{z}. \tag{57}$$

The controller $C(z,\rho)$ is said to belong to a controller class $\mathcal{C}$:

$$\mathcal{C} = \{C(z,\rho) : \rho \in \mathcal{P}\}, \tag{58}$$

where $\mathcal{P} \subseteq \mathbb{R}^p$ represents the set of admissible values for the parameter vector $\rho$. It is worth to mention that a necessary condition is that the design problem should present a unique solution, therefore the chosen controller class should represent the minimal possible number of parameters (minimal parametrization) (BAZANELLA; CAMPESTRINI; ECKHARD, 2012).

## 4.1 THE MODEL REFERENCE CONTROL

In the design paradigm of the Model Reference Control (MRC), the two-norm of the difference between the output signal $y(k)$ and a desired (represented by the subscript $d$) output signal $y_d(k)$ is considered:

$$J^{MR}(\rho) = ||y(k) - y_d(k)||_2^2, \tag{59}$$

where the desired output signal is obtained as

$$y_d(k) = T_d(z)r(k). \tag{60}$$

The $T_d(z)$ transfer function is defined as the *reference model*, such that its output, when a reference signal $r(k)$ is considered, results in the desired output $y_d(k)$. The cost function (59) can be rewritten as

$$J^{MR}(\rho) = ||(T(z,\rho) - T_d(z))r(k)||_2^2, \tag{61}$$

which has the minimum in zero. The MRC controller design, nevertheless, consists in finding the controller parameters $\rho$ such that (61) is minimized, i.e., making the closed-loop $T(z,\rho)$ as close as possible to the desired model $T_d(z)$. The solution is derived from expression (53) considering the reference model, as

$$T_d(z) = \frac{C_d(z)G(z)}{1 + C_d(z)G(z)}, \tag{62}$$

where the controller $C_d(z)$ is defined as the *ideal controller* since it is the exact solution of (61), and is given as

$$C_d(z) = \frac{T_d(z)}{G(z)(1 + T_d(z))}. \tag{63}$$

### 4.1.1 Choosing the reference model

The choice of $T_d(z)$ depends on the desired control objectives and must take some precautions into account, which are derived from expression (63). Regarding (63), some concerns about the properties of the controller $C_d(z)$ may arise. Thus, three main guidelines (BAZANELLA; CAMPESTRINI; ECKHARD, 2012) should be taken into account in order to avoid problems with the controller:

1. **Causality**: *the relative degree of the reference model $T_d(z)$ must be equal or greater than the relative degree* [1] *of the process $G(z)$*;

2. **Internal stability**: *in the presence of non-minimum phase zeros at the process $G(z)$, those must be included at the reference model $T_d(z)$*;

3. **Realistic ambition**: *the reference model $T_d(z)$ should be sufficiently close to what is possible to be achieved with the considered controller class $\mathcal{C}$.*

The *causality* guideline guarantees that the relative degree of $C_d(z)$ is non-negative, which makes the obtained controller causal and, therefore, possible to be applied in simulation or in practice. The first guideline requires the knowledge of the relative order of the process $G(z)$. Whilst, the *internal stability* guideline requires the knowledge of the NMP zeros of the process. The inclusion of the NMP zeros to the reference model allows for the cancellation of the poles outside of the unit circle that would appear in $C_d(z)$, considering expression (63). The presence of unstable poles in the controller is an undesired characteristic, since it leads to an internally unstable system (SKOGESTAD; POSTLETHWAITE, 2005).

Concerning performance, the *realistic ambition* guideline states that $T_d(z)$ should be chosen not requiring a too ambitious performance for the chosen controller structure - or class. E.g., a pure proportional controller itself would not be able to meet a null error in steady-state requirement for type-0 plants. Nevertheless, three control objectives can be highlighted:

1. Null error in steady-state. To be achieved, $T(e^{j\Omega_0}) = 1$ must be guaranteed for the frequency of interest $\Omega_0$, e.g., for dc-dc converters (KAZIMIERCZUK, 2008), aiming to control the output voltage, the frequency of interest is zero, whilst for inverters (ERICKSON; MAKSIMOVIC, 2001), the frequency of the output voltage is non-null - 50 Hz, 60 Hz;

2. Desired settling time for reference tracking performance analysis. It is imposed to $T_d(z)$ according to its dynamic behavior, defined mainly by its poles;

3. Overshoot criteria for a step reference. It is usually desirable to limit the overshoot for a step reference, avoiding practical problems at the controlled plant.

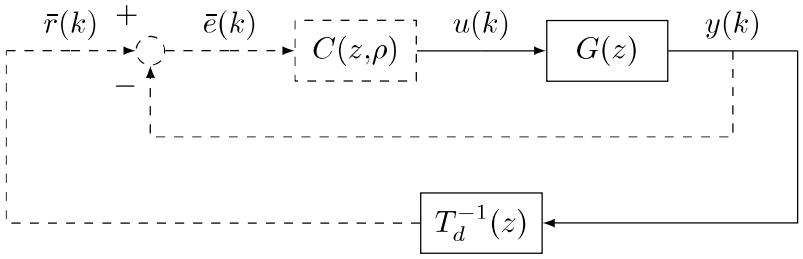
## 4.2 VIRTUAL REFERENCE FEEDBACK TUNING

The Virtual Reference Feedback Tuning (VRFT) (CAMPI; LECCHINI; SAVARESI, 2002) is a data driven controller design technique that solves the MRC problem. It requires only a single batch of input-output data (one-shot) to minimize its cost function, which is adapted from (61) and can be solved by the use of least squares when the controller is linearly parameterized as in (55). The result is the controller parameters $\rho$, considering a predefined controller class $\mathcal{C}$.

[1] Relative degree = (order of the denominator) - (order of the numerator).

The closed-loop block diagram for the VRFT controller design is illustrated in Figure 8, where disturbances and any type of noise are neglected. First, consider an experiment with the process $G(z)$, where an input signal $u(k)$, with a sufficient excitation order (LJUNG, 1999), is applied to the process, obtaining the output signal $y(k)$. The input and output signals are acquired, forming the data batch $\{u(k), y(k), k = 1...N\}$, where $N$ is the number of samples. A virtual error can be given as $\bar{e}(k) = \bar{r}(k) - y(k) = (T_d^{-1}(z) - 1)y(k)$, in which $\bar{r}(k)$ is the *virtual reference* signal obtained through the reference model $T_d(z)$ and the output signal $y(k)$. In summary, a controller $C(z,\rho) = \rho'\bar{C}(z)$ is considered satisfactory if it generates $u(k)$ when fed by $\bar{e}(k)$.

Figure 8 – Closed-loop block diagram for the VRFT controller design.




Source: (REMES et al., 2021b).

The MRC problem (61) is rewritten into a quadratic problem, resulting in the VRFT cost function

$$\underset{\rho}{\text{minimize}} \quad J^{VR}(\rho) \tag{64a}$$

$$J^{VR}(\rho) = ||u(k) - C(z,\rho)\bar{e}(k)||_2^2, \tag{64b}$$

which has the same minimum of (61) if the ideal controller $C_d(z)$ in (63) belongs to the controller class $\mathcal{C} = \{C(z,\rho), \rho \in \mathbb{R}^p\}$. The solution of (64) is obtained by least squares, as presented in Subsection 2.5.1, by rewriting (64b) as

$$J^{VR}(\rho) = ||u(k) - \rho'\phi(k)||_2^2, \tag{65}$$

where $\phi(k) = \bar{C}(z)\bar{e}(k)$, resulting in

$$\hat{\rho} = \left[\sum_{k=1}^{N} \phi(k)\phi'(k)\right]^{-1} \sum_{k=1}^{N} \phi(k)u(k). \tag{66}$$

In a practical case (66) would result in a biased estimation of $\hat{\rho}$, since the noise term $v(k)$ is present in the process. To cope with this, instrumental variables can be used to eliminate estimation bias (BAZANELLA; CAMPESTRINI; ECKHARD, 2012; CAMPI; LECCHINI; SAVARESI, 2002), requiring the acquisition of a second data batch. In practice, if there are no considerable memory restrictions, two identical input sequences can be used at the same

experiment. The acquired signals can be synced afterwards, resulting in two batches of input-output data from a single experiment.

In a practical case, the ideal controller is unknown - it is not possible to know, a priori, whether it belongs to the defined controller class. Thus, a filter $L(z)$ can be used to deal with the mismatch between the controller class and the ideal controller, by approximating the minima of $J^{VR}$ and $J^{MR}$. The magnitude of the filter $L(z)$ must satisfy the relation (BAZANELLA; CAMPESTRINI; ECKHARD, 2012)

$$|L(e^{j\Omega})|^2 = |T_d(e^{j\Omega})|^2|1-T_d(e^{j\Omega})|^2\frac{\Phi_r(e^{j\Omega})}{\Phi_u(e^{j\Omega})}, \quad \forall\Omega \in [-\pi,\pi], \tag{67}$$

where $x(e^{j\Omega})$, with $x$ being any signal, represents the Discrete Fourier Transform of $x(k)$, $\Phi_r(e^{j\Omega}), \Phi_u(e^{j\Omega})$ are, respectively, the power spectra of the signals $r(k), u(k)$. Any filter $L(z)$ that satisfies the magnitude restriction given in (67) approximates $J^{MR}$ and $J^{VR}$ minima. Since $\Phi_r(e^{j\Omega})$ is the spectra of the reference signal and $\Phi_u(e^{j\Omega})$ is the spectra of the input signal, choosing an input signal with the same waveform of the reference signal $r(k)$ that would be posteriorly applied to the plant, the relation $\Phi_r(e^{j\Omega})/\Phi_u(e^{j\Omega})$ can be considered unitary, allowing to obtain the filter - in this specific case - in the following implementable form:

$$L(z) = T_d(z)(1-T_d(z)). \tag{68}$$

Considering the filter $L(z)$ in the formulation, the cost function (65) becomes

$$J^{VR}(\rho) = ||L(z)(u(k)-\rho'\phi(k))||_2^2, \tag{69}$$

with the least squares solution given by

$$\hat{\rho} = \left[\sum_{k=1}^{N}\phi_L(k)\phi_L'(k)\right]^{-1}\sum_{k=1}^{N}\phi_L(k)u_L(k), \tag{70}$$

where $\phi_L(k) = L(z)\phi(k)$ and $u_L(k) = L(z)u(k)$ are the $\phi(k)$ and $u(k)$ signals filtered by $L(z)$.

At the presence of NMP zeros at the process $G(z)$, as aforementioned in Section 4.1, such zeros must be included in $T_d(z)$ to avoid poles outside the unit circle in the obtained controller $C_d(z)$. Notice that, even that the poles of the controller are fixed a priori due to the chosen controller structure, the VRFT problem seeks for a solution as close as possible to the ideal controller. Therefore, if the ideal controller is internally unstable, the identified controller will try to mimic its behavior, possibly leading to disastrous results, which highlights the importance of following guidelines at Subsection 4.1.1 to choose a reference model. The VRFT technique has as premise that the process $G(z)$ is completely unknown, except for its approximated relative order. Therefore, the location of the NMP zeros in the plant, if present, are also unknown. The inclusion of a flexible reference criterion at the VRFT cost function solves such problem and is presented in the next subsection.

### 4.2.1 Virtual Reference Feedback Tuning with flexible criterion

The VRFT with flexible criterion (CAMPESTRINI et al., 2011) extends the VRFT method to the case of non-minimum phase plants. In this case, the controller parameter $\rho$ is identified altogether with the zeros of the reference model, which ideally should be equal to the NMP zeros of the process $G(z)$.

The extension is done by considering a parameter vector $\eta \in \mathbb{R}^m$ that represents the gain and the $m-1$ zeros of $T_d(z,\eta)$. The reference model can, then, be written as

$$T_d(z,\eta) = \eta' F(z), \tag{71}$$

where $F(z)$ is a vector of transfer functions with dimension $m$. Firstly, $T_d(z)$ in (69) can be changed by $T_d(z,\eta)$, as

$$\tilde{J}^{VR}(\rho,\eta) = ||L(z)[u(k) - \rho'\bar{C}(z)(1 - T_d^{-1}(z,\eta))y(k)]||_2^2. \tag{72}$$

which can be multiplied by $T_d(z,\eta) = \eta' F(z)$ to avoid the parameter $\eta$ in the denominator, resulting in the flexible criterion

$$\tilde{J}^{VR}(\rho,\eta) = ||\eta' F(z)L(z)[u(k) - \rho'\bar{C}(z)(1 - \eta' F(z))y(k)]||_2^2. \tag{73}$$

Notice that (73) is bilinear in $\rho$ and $\eta$ simultaneously, allowing for the obtention of a solution through successive least squares.

The cost function, then, becomes

$$\underset{\rho,\eta}{\text{minimize}} \quad \tilde{J}^{VR}(\rho,\eta) \tag{74a}$$

$$\tilde{J}^{VR}(\rho,\eta) = ||\eta' F(z)L(z)[u(k) - \rho'\bar{C}(z)(1 - \eta' F(z))y(k)]||_2^2. \tag{74b}$$

which can be solved, at the *ith* iteration, as:

$$\begin{cases} \hat{\eta}_i = \arg\min_\eta \tilde{J}^{VR}(\hat{\rho}_{i-1},\eta); \\ \hat{\rho}_i = \arg\min_\rho \tilde{J}^{VR}(\rho,\hat{\eta}_i). \end{cases} \tag{75}$$

Notice that (75) requires an initial value of $\hat{\rho}$, named $\hat{\rho}_0$, or $\hat{\eta}$, called $\hat{\eta}_0$. Since the filter $L(z)$ becomes $L(z,\eta)$, it needs to be updated at each iteration according to $\hat{\eta}$, changing the magnitude requirement of (67) to

$$|L(e^{j\Omega},\eta)|^2 = |T_d(e^{j\Omega},\eta)|^2 |1 - T_d(e^{j\Omega},\eta)|^2 \frac{\Phi_r(e^{j\Omega})}{\Phi_u(e^{j\Omega})}, \quad \forall\Omega \in [-\pi,\pi]. \tag{76}$$

The solution for (75), at iteration $i$ for $\hat{\eta}_i$, is obtained as

$$\hat{\eta}_i = \left[\sum_{k=1}^{N} \phi_\eta(\hat{\rho}_{i-1},k)\phi_\eta'(\hat{\rho}_{i-1},k)\right]^{-1} \sum_{k=1}^{N} \phi_\eta(\hat{\rho}_{i-1},k)\tau_\eta(\hat{\rho}_{i-1},k), \tag{77}$$

with

$$\phi_\eta(\rho,k) = F(z)(u_L(k) + \rho'\bar{C}(z)y_L(k)), \tag{78}$$

$$\tau_\eta(\rho,k) = \rho'\bar{C}(z)y_L(k), \tag{79}$$

and for $\hat{\rho}_i$ as

$$\hat{\rho}_i = \left[\sum_{k=1}^{N} \phi_\rho(\hat{\eta}_{i-1},k)\phi'_\rho(\hat{\eta}_{i-1},k)\right]^{-1} \sum_{k=1}^{N} \phi_\rho(\hat{\eta}_{i-1},k)\tau_\rho(\hat{\eta}_{i-1},k), \tag{80}$$

where

$$\phi_\rho(\eta,k) = \bar{C}(z)(1 - T_d(z,\eta))y_L(k), \tag{81}$$

$$\tau_\rho(\eta,k) = T_d(z,\eta)u_L(k), \tag{82}$$

with $u_L(k) = L(z,\eta)u(k)$ and $y_L(k) = L(z,\eta)y(k)$. Notice that the solutions presented in (77) and (80) still result in a biased estimation in the presence of noise $v(k)$. Nevertheless, instrumental variables can be used the same way as aforementioned. It is worth to mention that, although the solution for the VRFT with flexible criterion is iterative, it still only requires a single batch of data.

## 4.3 FINAL CONSIDERATIONS

This chapter presented the Model Reference Control, which is used as a base for the Virtual Reference Feedback Tuning method, also addressed here. The VRFT is detailed for the cases where NMP zeros are present or not. A filter is used to suppress the mismatch between the chosen controller class and the required controller to achieve the reference model dynamic behavior in closed-loop. Both VRFT solutions rely only on a single batch of data and can be obtained through least squares.

# 5 SWARM INTELLIGENCE ALGORITHMS

Metaheuristics are optimization methods mainly used for complex problems that find an approximate solution, ideally close to the optimal solution (TALBI, 2009). Swarm intelligence algorithms are metaheursitics that mimic the social behavior of animals. They are stochastic algorithms, iterative, and population-based - i.e., they have several search agents, which enhance the diversity of the searching.

The NFL theorems (WOLPERT; MACREADY, 1997) state that a metaheuristic optimization algorithm that does well on average for a class of problems, will do worse on average over other class of problems. There is a need, thus, to test more than one algorithm for the proposed problem. Within all the metaheuristic classes of algorithms, the swarm intelligence algorithms are chosen to be used since they present the least number of hyperparameters within metaheuristics (NADIMI-SHAHRAKI; TAGHIAN; MIRJALILI, 2021). In this chapter, four swarm intelligence algorithms are presented and detailed according to the following sections, for the reasoning described in Section 2.5.2: Particle Swarm Optimization (PSO) (KENNEDY; EBERHART, 1995); Artificial Bee Colony (ABC) (KARABOGA; BASTURK, 2007); Grey Wolf Optimizer (GWO) (MIRJALILI; MIRJALILI; LEWIS, 2014); and Improved Grey Wolf Optimizer (I-GWO) (NADIMI-SHAHRAKI; TAGHIAN; MIRJALILI, 2021).

## 5.1 PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (KENNEDY; EBERHART, 1995) is a stochastic particle system in which the position and velocity (social behavior) of each particle is adjusted at each iteration. The population number is static during the optimization procedure. The PSO is said to mimic the buzzing, flocking, and schooling behavior of bees, birds, and fish (TALBI, 2009; DU; SWAMY, 2016; KENNEDY; EBERHART, 1995).

In the basic model of the PSO, $\ell$ particles that start at a random location with random velocity compose the swarm within a search space of dimension $D$. The main characteristics of a particle $i$ are its position ($\overrightarrow{X}_i = \{X_{i1}, X_{i2}, ..., X_{iD}\}$) and its velocity ($\overrightarrow{V}_i = \{V_{i1}, V_{i2}, ..., V_{iD}\}$). Each particle is considered to be a possible solution for the problem. The velocity of a particle is calculated as $\overrightarrow{V}_i(n) = \overrightarrow{X}_i(n) - \overrightarrow{X}_i(n-1)$, where $\overrightarrow{X}_i(n-1)$ and $\overrightarrow{X}_i(n)$ are the locations of the particle $i$ in space at instant $n-1$ and $n$, respectively. The best solution ever found by a particle ($pbest_i$), i.e., locally, is stored in a vector $\overrightarrow{P}_i = \{P_{i1}, P_{i2}, ..., P_{iD}\}$. The best *global* solution ever found by the swarm (*gbest*) is stored as $\overrightarrow{G} = \{G_1, G_2, ..., G_D\}$.

The new values of position and velocity at each iteration depend on their previous values, as well as on their neighborhood. The update of velocity at each iteration is given by:

$$\overrightarrow{V}_i(n) = w_1 \overrightarrow{V}_i(n-1) + w_2 C_1 (\overrightarrow{P}_i - \overrightarrow{X}_i(n-1)) + w_3 C_2 (\overrightarrow{G} - \overrightarrow{X}_i(n-1)), \tag{83}$$

where $w_1$ represents inertia, i.e., a weight that controls the impact of previous velocity ($\overrightarrow{V}(n-1)$) at the current velocity ($\overrightarrow{V}(n)$), $w_2$ and $w_3$ are random variables such that $w_2, w_3 \sim U(0,1)$,

constant $C_1$ is the cognitive learning factor (the attraction of a particle to its own success), and constant $C_2$ is the social learning factor (attraction of a particle to the success of its neighbors)

After updating the velocity, the position of each particle is updated according to

$$\overrightarrow{X}_i(n) = \overrightarrow{X}_i(n-1) + \overrightarrow{V}_i(n). \tag{84}$$

Any particle might generate a solution with less fitness at each iteration. Thus, the current best solution is checked and updated as

$$\begin{cases} \overrightarrow{P}_i(n+1) = \overrightarrow{X}_i(n), & \text{if } f(\overrightarrow{X}_i) < \overrightarrow{P}_i(n); \\ \overrightarrow{P}_i(n+1) = \overrightarrow{P}_i(n), & otherwise, \end{cases} \tag{85}$$

where $f(\cdot)$ is the mapping of the cost function that the algorithm is optimizing. The update of the global solution, i.e., the best solution found by all the swarm within all iterations, is done accordingly:

$$\begin{cases} \overrightarrow{G}(n+1) = \overrightarrow{X_i}(n), & \text{if } f(\overrightarrow{X}_i(n)) < G(n); \\ \overrightarrow{G}(n+1) = \overrightarrow{G}(n), & otherwise. \end{cases} \tag{86}$$

The point where a particle will cycle around is given by the weighted average of the best local position visited by each particle $\overrightarrow{P}_i = \{P_{i1}, P_{i2}, ..., P_{iD}\}$ and the best global position visited by the swarm $\overrightarrow{G} = \{G_1, G_2, ..., G_D\}$:

$$\frac{w_2 \overrightarrow{P}_i + w_3 \overrightarrow{G}}{w_2 + w_3}. \tag{87}$$

A stopping criterion is responsible for stopping the execution of the PSO, which is usually: an external function; a minimum cost to be obtained; or a maximum number of iterations. The best global solution $\overrightarrow{G}(n+1)$ at the iteration when the algorithm stops is taken as the solution of the optimization problem.

## 5.2 ARTIFICIAL BEE COLONY

The artificial bee colony algorithm simulates the foraging behavior of honey bees. The bees conduct a local search at each iteration for food (nectar) sources. Each food source (position vector) is a possible solution based on its availability of nectar (fitness). The maximum number of food sources are limited, and for each food source, there is an employed bee looking for nectar (DU; SWAMY, 2016; KARABOGA; BASTURK, 2007).

The bees are classified in three main types: i) employed bees, which are bees that look for food sources in the neighborhood of a current food source that is being explored; ii) onlooker bees that receive information of all food sources and select the best ones; and iii) scouts, which are bees that look for new food sources when a current food source cannot be improved anymore.

The execution of the algorithm is divided in four phases:

1. **Initialization phase:** $\overrightarrow{X}_i = \{X_{i1}, X_{i2}, ..., X_{iD}\}$ are the $\ell$ food sources with dimension $D$, in which $i = 1, ..., \ell$, and are initialized according to

$$\overrightarrow{X}_i(0) = l_b + r_b(u_b - l_b), \tag{88}$$

where $l_b$ is the lower search bound, $u_b$ is the upper search bound, and $r_b$ is a random number such that $r_b \sim U(0,1)$.

2. **Employed bees phase:** employed bees are bees that are collecting nectar from a food source and, simultaneously, looking for better food sources on its neighborhood, with position updated at each iteration according to

$$\overrightarrow{X}_i(n+1) = \overrightarrow{X}_i(n) + r_a(\overrightarrow{X}_i(n) - \overrightarrow{X}_r(n)), \tag{89}$$

where $\overrightarrow{X}_i(n+1)$ is the food source at iteration $n+1$ and $\overrightarrow{X}_i(n)$ is the food source at iteration $n$, $r_a$ is a random number such that $r_a \sim U(-a,a)$ where $a$ is named acceleration coefficient, and $\overrightarrow{X}_r(n)$ is an existing food source from iteration $n$ that is randomly selected. The food source at iteration $n$ and the new found food source at iteration $n+1$ are compared. A greedy selection - that compares food sources and choose the best one in terms of food (cost) - decides which food source will be considered at the next iteration. The information of the position of food sources and the amount of nectar (fitness) is shared with the onlooker bees, which are waiting in the hive;

3. **Onlooker bees phase:** when onlooker bees receive information from the employed bees about the food sources positions and fitness, a food source is selected among all $L$ food sources with probability $P_i$, defined as

$$P_i = \frac{f(\overrightarrow{X}_i)}{\sum_{j=1}^{L} f(\overrightarrow{X}_j)}, \tag{90}$$

where $f(\cdot)$ is the mapping of the cost function to be minimized, which evaluates the fitness of a solution $\overrightarrow{X}_i$. When a food source is selected, a neighbor food source is determined by expression (89). A greedy selection is applied to both food sources, maintaining the one with more nectar (less fitness);

4. **Scout bees phase:** when a food source (solution) cannot be improved anymore for a determined number of iterations, or the limit $L$ - the maximum allowed number of food sources - is reached, the employed bees abandon the food source and become scouts, which randomly choose a new food source as done in the initialization phase (88).

The bees' phases (2, 3, and 4) are repeated until a stopping criterion is met, i.e., maximum number of iterations, cost threshold obtained, etc. At the end of the execution, the food source with more nectar (less fitness) is considered as the global solution and, therefore, the solution of the optimization problem.

## 5.3 GREY WOLF OPTIMIZER

The GWO algorithm is based on grey wolves' social hierarchy and hunting behavior. In a pack, there are four types of wolves: i) alphas, which are the leaders, responsible for making decisions; ii) betas, subordinates to the alphas that help in decision-making and other pack activities; iii) deltas, which represent scouts, sentinels, elders, hunters, and caretakers; and iv) omegas, which compose the rest of the pack and submit to higher ranking wolves (MIRJALILI; MIRJALILI; LEWIS, 2014).

All $\ell$ wolves are initialized randomly (with an uniform distribution) within the whole search space. The three best solutions (wolves positions) of the optimization procedure are defined by the position vectors alpha ($\overrightarrow{X}_{\alpha}$), beta ($\overrightarrow{X}_{\beta}$), and delta ($\overrightarrow{X}_{\delta}$). The rest of the wolves are assumed to be omega, with position $\overrightarrow{X}_{\omega,k}$, where $k$ represent a specific wolf among the omegas with $k \in \{1...\ell-3\}$, and will follow the three best solutions, according to the mean value of those.

Grey wolves present an encircling behavior to its prey when hunting, which can be mathematically modeled as follows: the displacement of the $\alpha$, $\beta$, and $\delta$ wolves is defined as

$$\overrightarrow{D}_p(n) = |\overrightarrow{C}\,\overrightarrow{X}_p(n) - \overrightarrow{X}_i(n)|, \tag{91}$$

where $\overrightarrow{C} = 2\overrightarrow{r}_1$, $r_1 \in \mathbb{R}^p$ is a random vector such that $r_1 \sim U(0,1)$, and $p \in \{\alpha,\beta,\delta\}$, and $\overrightarrow{X}_i(n)$ is the position a wolf $i \in \{\alpha;\beta;\delta;\omega,k\}$ that is being updated. The weighted position of the three wolves with the best solution in terms of cost at iteration $n$, i.e., $\overrightarrow{X}_p$, $p = \{\alpha,\beta,\delta\}$, are calculated as

$$\overrightarrow{X}_{w\alpha} = \overrightarrow{X}_{\alpha}(n) - \overrightarrow{A}_1\overrightarrow{D}_{\alpha}(n), \tag{92}$$

$$\overrightarrow{X}_{w\beta} = \overrightarrow{X}_{\beta}(n) - \overrightarrow{A}_2\overrightarrow{D}_{\beta}(n), \tag{93}$$

and

$$\overrightarrow{X}_{w\delta} = \overrightarrow{X}_{\delta}(n) - \overrightarrow{A}_3\overrightarrow{D}_{\delta}(n), \tag{94}$$

with coefficient vectors $\overrightarrow{A}_i$, $i = 1,2,3$ given by

$$\overrightarrow{A}_i = 2\overrightarrow{a}\,\overrightarrow{r}_2 - \overrightarrow{a}, \tag{95}$$

where $\overrightarrow{a}$ is linearly decreased from 2 to 0 over iterations in order to mathematically model the approaching to the prey during the encircle, and $r_2 \in \mathbb{R}^p$ is a random vector such that $r_2 \sim U(0,1)$. The position of all wolves in the pack, for the next instant $(n+1)$, is calculated as

$$\overrightarrow{X}_i(n+1) = \frac{\overrightarrow{X}_{w\alpha} + \overrightarrow{X}_{w\beta} + \overrightarrow{X}_{w\delta}}{3}. \tag{96}$$

Since the position of all wolves at iteration $(n+1)$ are updated, the three wolves with lower fitness at this iteration are now considered to be the alpha, beta, and delta, whilst the rest is taken

as omega. Also, notice that the position of the *prey* is mathematically described as the average of the three best solutions. From the encircling behavior, the prey should be approximately in the middle of the three "best" wolves at the iteration in question.

The optimization procedure continues until a user defined stopping criterion is met. The best found solution at the stopping iteration, the position of the alpha $\overrightarrow{X}_{\alpha}$, is considered the solution of the minimization procedure.

## 5.4 IMPROVED GREY WOLF OPTIMIZER

The GWO algorithm, although effective, has shown three main problems that can be observed in literature (HEIDARI; PAHLAVANI, 2017; LONG et al., 2018; LU; GAO; YI, 2018; TU; CHEN; LIU, 2019):

1. Lack of population diversity, meaning that the wolves in GWO only follow hierarchy, not being influenced by its close neighbors;
2. Imbalance between the exploitation and exploration;
3. Premature convergence.

As an extension to suppress such problems, the *Improved* Grey Wolf Optimizer (NADIMI-SHAHRAKI; TAGHIAN; MIRJALILI, 2021) presents a change in the search strategy of the main GWO algorithm, composed of three phases: initialization; movement; selection and update. The three phases of the I-GWO algorithm are described below:

1. **Initialization:** $\ell$ grey wolves are randomly distributed throughout the search space, with lower bound $l_j$ and upper bound $u_j$, as

$$\overrightarrow{X}_{ij} = l_j + r_j(u_j - l_j), \quad i \in [1,\ell], \quad j \in [1,D], \tag{97}$$

   where $D$ is the problem's dimension, $\overrightarrow{X}_{ij}$ forms a population matrix of grey wolves $Pop$, and $r_j$ is a random variable such that $r_j \sim U(0,1)$;

2. **Movement:** one of the improvements of the I-GWO algorithm in relation to the GWO algorithm is the inclusion of individual hunting. This is done through a strategy named Dimension Learning-based Hunting (DLH). A radius is defined as the Euclidean distance between the current position $\overrightarrow{X}_i(n)$ and the GWO candidate position $\overrightarrow{X}_{i,GWO}(n+1)$, which is calculated the exact same way as in GWO via (96), as

$$\overrightarrow{R}_i(n) = ||\overrightarrow{X}_i(n) - \overrightarrow{X}_{i,GWO}(n+1)||_2^2. \tag{98}$$

   The neighbors of $\overrightarrow{X}_i(n)$, denoted by $\overrightarrow{N}_i(n)$, can be defined as

$$\overrightarrow{N}_i(n) = \left\{ \overrightarrow{X}_j(n) \mid ||\overrightarrow{X}_i(n) - \overrightarrow{X}_j(n)|| \leq \overrightarrow{R}_i(n), \overrightarrow{X}_j(n) \in Pop \right\} \; i \in [1,\ell], \; j \in [1,D]. \tag{99}$$

Multi-neighbor learning (DLH) is performed, resulting in the DLH candidate solution

$$\overrightarrow{X}_{i,DLH}(n+1) = \overrightarrow{X}_i(n) + r_i(\overrightarrow{X}_n(n) - \overrightarrow{X}_r(n)), \tag{100}$$

where $\overrightarrow{X}_n(n)$ is a random neighbor within neighborhood $\overrightarrow{N}_i(n)$, $\overrightarrow{X}_r(n)$ is a random grey wolf from the population matrix $Pop$, and $r_i$ a random vector such that $r_i \sim U(0,1)$;

3. **Selection and update:** at the last phase, the fitness value of all solutions (wolves) are compared and selected, if the fitness is less than the lowest fitness obtained already, according to:

$$\overrightarrow{X}_i(n+1) = \begin{cases} \overrightarrow{X}_{i,GWO}(n+1), & \text{if } f(\overrightarrow{X}_{i,GWO}(n+1)) < f(\overrightarrow{X}_{i,DLH}(n+1)); \\ \overrightarrow{X}_{i,DLH}(n+1), & \text{otherwise.} \end{cases} \tag{101}$$

The best solution ever found is updated if $\overrightarrow{X}_i(n)$ has a fitness value that is greater than the new possible best solution, $\overrightarrow{X}_i(n+1)$, evaluated by $f(\overrightarrow{X}_i(n+1))$, where $f(\cdot)$ is the mapping of the cost function to be minimized.

An user defined stopping criterion is responsible for stopping the execution of the I-GWO algorithm, may it be a maximum number of iterations or a minimum required fitness value. The position of the alpha wolf, $\overrightarrow{X}_\alpha$, at the last executed iteration, is considered to be the solution of the optimization procedure.

## 5.5 FINAL CONSIDERATIONS

The four swarm intelligence algorithms considered in this work were detailed in this chapter. PSO regards the movement of particles within a search space, imitating the flocking of birds (or similar animal behavior). The ABC algorithm mimics the foraging behavior of honey bees. GWO and I-GWO relates to the hunting of grey wolves, with the I-GWO presenting certain improvements to the algorithm of the GWO in order to avoid local minima.

A relevant characteristic of the algorithms to the user is the number of hyperparameters that are needed to be set, arbitrarily or not. Table 5 shows the parameters that shall be chosen by the user for each mentioned algorithm. As it can be seen in Table 5, the GWO and I-GWO present the least number of hyperparameters, whilst PSO has the most.

Table 5 – Hyperparameters of the considered swarm intelligence algorithms.

| Algorithm | Parameter settings |
|---|---|
| PSO | Maximum number of iterations |
| | Number of agents ($\ell$) |
| | Cognitive learning factor ($C_1$) |
| | Social learning factor ($C_2$) |
| | Inertia range (range of $w_1$) |
| ABC | Maximum number of iterations |
| | Number of agents ($\ell$) |
| | Limit of food sources ($L$) |
| | Acceleration coefficient ($a$) |
| GWO | Maximum number of iterations |
| | Number of agents ($\ell$) |
| I-GWO | Maximum number of iterations |
| | Number of agents ($\ell$) |

Source: the author.

# 6 THE PROPOSED METHOD: VIRTUAL REFERENCE FEEDBACK TUNING WITH ROBUSTNESS CONSTRAINT

This chapter presents the proposed method of this thesis: a two-step procedure, where the first step follows the design of a controller by the VRFT method as commented in Chapter 4. The second step consists on the application of swarm intelligence algorithms to increase the robustness of the closed-loop system by finding new controller parameters, based on the solution of step one.

## 6.1 ESTIMATION OF ROBUSTNESS INDEX WITH A SINGLE BATCH OF DATA

The method to estimate the $\mathcal{H}_\infty$ norm of $S(z,\rho)$, described in Section 3.2, considers the closed-loop data obtained with the already designed controller $C(z,\hat{\rho})$, providing the robustness index of the controlled system. Such $M_S$ estimation needs to be extended to a case where the norm of $S(z,\hat{\rho})$ is dependent of $C(z,\hat{\rho})$, in order to maintain the one-shot characteristic during the metaheuristic optimization procedure. The same data that was acquired for the VRFT design is used to estimate the robustness index.

Considering the system described in Subsection 2.3, and that the controller is dependent of the parameter vector $\rho$, its sensitivity transfer function $S(z,\rho)$ is obtained from

$$T(z,\rho) = \frac{C(z,\rho)G(z)}{1+C(z,\rho)G(z)} \tag{102}$$

and

$$S(z,\rho) + T(z,\rho) = 1, \tag{103}$$

resulting in

$$S(z,\rho) = \frac{1}{1+C(z,\rho)G(z)}, \tag{104}$$

which can be rewritten as

$$1 + C(z,\rho)G(z) = S^{-1}(z,\rho). \tag{105}$$

Assuming that $u(k)$ is sufficiently informative to capture all relevant characteristics of $S(z,\rho)$, and multiplying both sides of (105) by $u(k)$, it is obtained that

$$u(k) + C(z,\rho)G(z)u(k) = S^{-1}(z,\rho)u(k). \tag{106}$$

It is known that $G(z)u(k) = y(k)$. Substituting such relation in (106):

$$u(k) + C(z,\rho)y(k) = S^{-1}(z,\rho)u(k). \tag{107}$$

Finally, the signals

$$\xi(k) = u(k), \quad \zeta(k) = u(k) + C(z,\rho)y(k), \tag{108}$$

can be defined such that $S(z,\rho)\zeta(k) = \xi(k)$, i.e., when a signal $\zeta(k)$ formed by $u(k)+C(z,\rho)y(k)$ is applied to $S(z,\rho)$, an output $\xi(k) = u(k)$ is obtained. The impulse response of $S(z,\rho)$, and consequently the $\mathcal{H}_\infty$ norm (i.e., the robustness index), can be estimated with the data set $\{\xi(k), \zeta(k), k = 1...N\}$, through the method presented in Chapter 3.

## 6.2 DESCRIPTION OF THE METHOD

The Virtual Reference Feedback Tuning with robustness constraint can be divided in two main steps. The first step of the proposed method regards the design of a controller using the VRFT as presented in Chapter 4. It is assumed that a batch of input-output data, used in step 1 to design the controller, is available. The second step accounts for the use of swarm intelligence algorithms to minimize a modified VRFT cost function, with an $\mathcal{H}_\infty$ robustness constraint, as presented below. Figure 9, at the end of the chapter, summarizes the method in a flowchart form and can be used to increase the comprehension of the method description in the next subsection.

### 6.2.1 Robustness constraint to the VRFT formulation

The VRFT is based on the optimization of a cost function, as described in Chapter 4. An extension to the formulation, regarding a robustness constraint - the $\mathcal{H}_\infty$ norm of $S(z,\rho)$ - at such cost function (69) leads to a new optimization problem:

$$\begin{aligned} \underset{\rho}{\text{minimize}} \quad & J^{VR}(\rho) \\ \text{subject to} \quad & \hat{M}_S(\rho) \leq M_{Sd}, \end{aligned} \tag{109}$$

where $\hat{M}_S(\rho)$ is the estimated $||S(z,\rho)||_\infty$ and $M_{Sd}$ is the maximum allowed value. The constraint can be applied to the cost function in the form of a penalty or a barrier, as mentioned in Chapter 2, Section 2.5. It is chosen to apply the constraint in the form of a penalty, since the resultant cost function does not present any, even that minimal, constraint, which is not the case for barrier methods. The resultant *Swarm Intelligence* optimization cost function is:

$$\underset{\rho}{\text{minimize}} \quad J^{SI}(\rho) \tag{110a}$$

$$J^{SI}(\rho) = J^{VR}(\rho) + cH(\rho) \tag{110b}$$

where $J^{VR}(\rho) = ||L(z)(u(k) - C(z,\rho)(T_d^{-1}(z) - 1)y(k))||_2^2$ and the penalty term can be applied as

$$H(\rho) = \frac{1}{2}(max[0, \hat{M}_S(\rho) - M_{Sd}])^2, \tag{111}$$

as a function of $\rho$, accounting for the estimated ($\hat{M}_S$) and desired ($M_{Sd}$) maximum value of the robustness index. Such index is estimated at each iteration of the swarm algorithm optimization, according to the method presented in Chapter 3 with the extension at Section 6.1.

The considered search space is $O \in [l_b, u_b], l_b, u_b \in \mathbb{R}$, where $l_b$ is the lower bound and $u_b$ is the upper bound. In order to accelerate convergence, the initialization of the swarm agents may inherit the first step solution (the VRFT solution), which is denoted as $\rho_0 \in \mathbb{R}^p$, as a central point of spawn, as

$$\overrightarrow{X}_i(0) = R \cdot \overrightarrow{X}(0) + \rho_0, \quad i \in [1, \ell] \tag{112}$$

where $\overrightarrow{X}(0) \in \mathbb{R}^p$ is a random position vector such that $\overrightarrow{X}(0) \sim U(0,1)$ and the initial population spawn radius can be defined as

$$R = \frac{|l_b| + |u_b|}{2}, \tag{113}$$

since it reduces the initial spawn radius in half which accelerates the convergence of the meta-heuristic optimization algorithm. In other words, it is assumed that the robust solution is possible inside a disk, centered in $\rho_0$, with radius $R$. In order to use (113), it is considered that the designer will have a *realistic ambition* when selecting the value of $M_{Sd}$, since, e.g., a too low value of desired maximum robustness index for a closed-loop system with a very high robustness index may result in a very poor reference tracking performance. If the chosen $M_{Sd}$ is not too far from the actual $M_S$ of the system with the VRFT-designed controller, the robust-constrained solution should not be too far from the VRFT-obtained solution, which would be located inside the biased spawn radius (113).

An inherent step of the proposed method is to collect input-output data from the process. Remember that, for data to be sufficiently informative, the input signal used as excitation to the system must be persistently exciting of high order (LJUNG, 1999). After the acquisition of data, the two steps of the proposed method can be applied:

1. Use the VRFT to design a controller for the process. Use a flexible reference model if the plant is NMP, as done in the base literature (CAMPESTRINI et al., 2011). Check the obtained robustness index, proceed to the second step if it does not satisfy $\hat{M}_S < M_{Sd}$. Such main step can be divided into the following specific steps:

    - acquire a data set $\{u(k), y(k), k = 1...N\}$ from the closed-loop system with an initial stabilizing controller;
    - use the data set to design a controller using the VRFT method, as detailed in Chapter 4. Controller parameters $\hat{\rho}$ are obtained after the minimization procedure of the VRFT method. In the NMP case, parameters for the reference model $\hat{\eta}$ are also obtained;
    - estimate the robustness index according to the method in Section 6.1. If $\hat{M}_S > M_{Sd}$ proceed to the second step, else, use the VRFT-obtained controller with no further modification.

2. Apply a swarm intelligence algorithm considering the optimization problem described in (110) according to a desired value of $M_S$, with restriction applied in the form of a penalty as (111), with initial spawn of agents following the recommendation of (112). Such main step can be divided into the following specific steps:

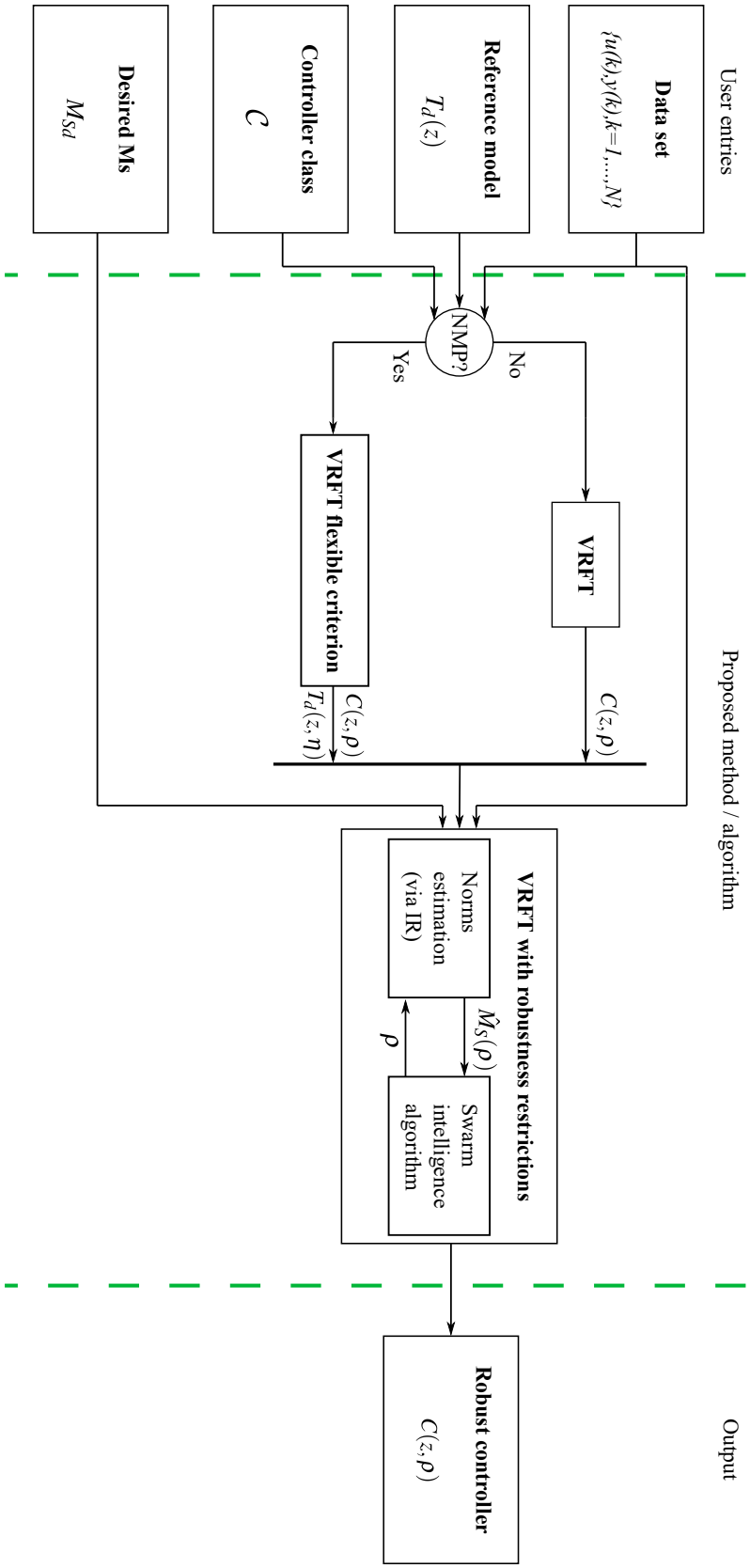
    - implement the VRFT cost function with the penalty as in (111) regarding the desired maximum value of $M_S$;

- change the initialization procedure of the chosen swarm intelligence algorithm to consider a center spawn $\rho_0$, i.e., the VRFT-obtained solution at the first step, and a spawn radius as suggested in (112) to accelerate convergence;
- run the algorithm and obtain controller parameters that satisfy the robustness restriction.

## 6.3 FINAL CONSIDERATIONS

This chapter formulated the proposed method according to the previously mentioned subjects and techniques. It consists of a two-step method. The first step regards the design of a controller using the VRFT, and the second addresses the application of a swarm intelligence algorithm to a cost function based on the VRFT with the inclusion of a robustness constraint in the form of a penalty.

Figure 9 – Flowchart diagram summarizing the proposed robust solution for VRFT using swarm intelligence algorithms.




Source: the author.

# 7 VALIDATION RESULTS

This chapter validates and illustrates the proposed method with two examples inspired on the structure of dc-dc converters. The application of the method is done as suggested in Chapter 6, considering the swarm intelligence algorithms presented in Chapter 5. The obtained results are compared in terms of: i) fitness value obtained considering the best solution of the swarm (best fitness); ii) $\mathcal{H}_\infty$ norm of $S(z,\rho)$, $M_S$, obtained with the best solution; and iii) convergence speed. Notice that the system model is only used in simulation for data acquisition, whilst its model knowledge is *neglected at all stages of the design procedure*, keeping a pure data-driven approach.

Most of the results in this chapter are analyzed in terms of box plots, which are graphical representations of data through quartiles, presenting information regarding their median, maxiximum and minimum values, as well as their dispersion - mentioned sometimes as the standard deviation. Outliers, which differ significantly from the rest of the database, are also indicated in the box plots. Each box plot is followed by a table with the most relevant values regarding the graphical representation.

## 7.1 EXAMPLE 1: A SECOND-ORDER PLANT

The first system to be considered is

$$G_1(z) = \frac{-0.05(z-1.4)}{z^2-1.7z+0.7325} \tag{114}$$

which is similar to the structure of a discrete-time model of a Boost/Buck-Boost type converter operating in Continuous Conduction Mode (CCM) when regarding the output voltage by duty cycle transfer function (ERICKSON; MAKSIMOVIC, 2001), containing two complex conjugate poles and a zero. The model parameters were chosen such that the robustness index, obtained after the VRFT design, was higher than 2 (absolute). As the zero is outside of the unit circle, the plant can be classified as Non-Minimum Phase (NMP), requiring for the use of a flexible criterion at the VRFT design procedure (CAMPESTRINI et al., 2011). The plant $G_1(z)$ is used to simulate a real system, from which data is collected in order to design the controller. Notice that the system model $G_1(z)$ is assumed unknown for controller design, so there is no prior knowledge about its zero or any other parameter.

### 7.1.1 Data collection

The data $\{u(k), y(k), k = 1...N\}$ for the VRFT design is acquired in closed-loop with an initial proportional stabilizing controller (REMES et al., 2021a). The small gain theorem (SKOGESTAD; POSTLETHWAITE, 2005) states that a proportional controller is stabilizing if

$$k_p < \frac{1}{||G||_\infty}. \tag{115}$$

Therefore, the controller is obtained as

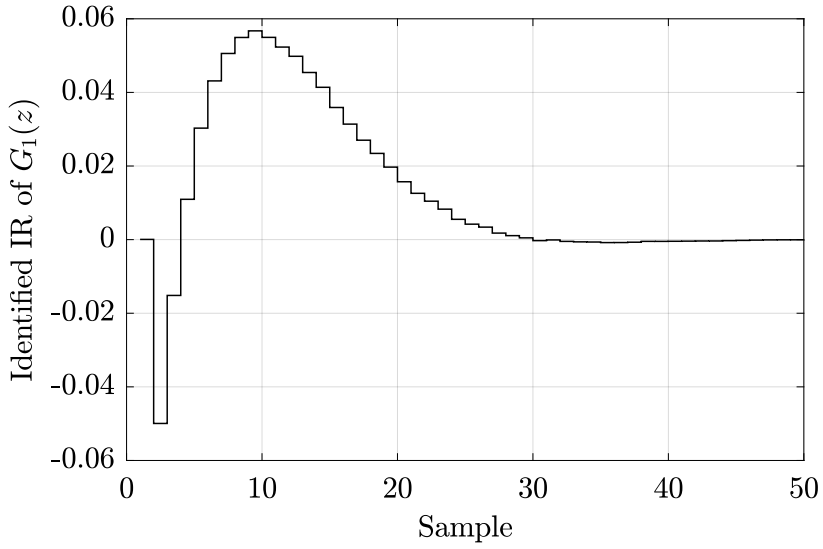
$$k_p = \frac{0.5}{||G_1(z)||_\infty} = 0.8039. \tag{116}$$

The $\mathcal{H}_\infty$ norm of $G_1(z)$ is obtained with the estimation method proposed in Chapter 3.

For the VRFT design, a PRBS with $N = 2000$ samples is applied as reference to the plant, in closed-loop with the proportional controller $k_p$ obtained in (116). Additive white Gaussian noise with a Signal-to-Noise Ratio (SNR) of 20 dB was added to the system at the output and fed back to represent measurement noise.

In order to investigate the presence of NMP zeros at the plant, the IR can be analyzed. In NMP systems the impulse response initially moves downwards (BRUNTON; KUTZ, 2019) - the "opposite" direction, see Figure 10 - which can be verified graphically. Observe that the IR can be estimated with the acquired data set for the VRFT design (step 1 of the proposed algorithm), $\{u(k), y(k), k = 1...N\}$, being available to be analyzed by the user or by a simple algorithm. The IR, which is presented in Figure 10, can be identified with regularized impulse response estimation algorithms, as described in Chapter 3 and available in programming libraries (FIORIO; REMES; NOVAES, 2021; THE MATHWORKS INC., 2021; YERRAMILLI; TANGIRALA, 2017). It can be noted that the first non-null element of the IR has a negative value, which visually means that the IR initially goes "downwards", indicating the presence of at least one NMP zero. Thus, the VRFT must be approached with a flexible reference criterion (CAMPESTRINI et al., 2011), as detailed in Subsection 4.2.1.

Figure 10 – Identified impulse response of $G_1(z)$.




### 7.1.2 Step 1 - VRFT with flexible criterion

This subsection has the objective of designing a controller using VRFT with flexible criterion, according to the first step of the proposed method. This controller shall be used as the

initial controller for the second step. In order to design such controller, the following requirements are considered:

1. Null error in steady state;
2. Settling time of approximately 2.5 times faster than the settling time in closed-loop with the stabilizing controller $k_p$;
3. Null overshoot for a step reference.

Notice that such requirements, mainly the 2.5 times faster settling time, relates to a too ambitious performance, as mentioned in the last chapters. This type of situation may happen in practice, e.g., if the designer is inexperienced or if the designer has few empirical knowledge about the plant to be controlled. This poor choice of control requirements is a characteristic that might result in poor robustness with the VRFT-designed controller.

The null error in steady state can be guaranteed by choosing a reference model such that $T_d(e^{j\Omega_0}) = 1$ at the frequency of interest $\Omega_0$. Considering that the plant is based in the structure of a dc-dc converter, the frequency of interest is the frequency of a dc signal, which is zero. Than, $\Omega_0 = 0$, which gives the criteria

$$T_d(1) = 1. \tag{117}$$

The desired settling time is obtained by the position of the poles of the reference model (REMES et al., 2021a). Finally, a requirement for null overshoot is to have real poles at the model, resulting in

$$T_d(z, \hat{\eta}_0) = \frac{-21(z-1.01)}{(z-0.7)(z-0.3)}, \tag{118}$$

Notice that an initial zero is included at 1.01, as suggested in the basis literature (REMES et al., 2021a), so that the VRFT with flexible reference may identify correctly the NMP zero of the plant. The controller class, chosen to be used in this example, is the PID class of controllers,

$$\bar{C}(z) = \begin{bmatrix} 1 & \dfrac{z}{z-1} & \dfrac{z-1}{z} \end{bmatrix}', \tag{119}$$

since it is a widely used class of controllers, and at the same time, it represents a low order controller, with fewer hardware requirements to be applied in practice if compared to higher order controller classes.

Through the VRFT method with flexible criterion, solving the problem (74) through successive least squares as indicated in Chapter 4, the solution pair $\hat{\eta}, \hat{\rho}$ is obtained as

$$\hat{\eta} = \begin{bmatrix} -0.4793 & 0.6377 \end{bmatrix}' \tag{120a}$$

$$\hat{\rho} = \begin{bmatrix} 1.1246 & 0.3124 & 6.9713 \end{bmatrix}', \tag{120b}$$

resulting in the reference model $T_d(z,\eta) = \eta' F(z)$ and in the controller $C(z,\rho) = \rho' \bar{C}(z)$:

$$T_d(z,\hat{\eta}) = \hat{\eta} F(z) = \frac{-0.6899(z-1.33)}{(z-0.7)(z-0.2401)}, \tag{121a}$$

$$C(z,\hat{\rho}) = \hat{\rho}' \bar{C}(z) = \frac{8.4083(z^2-1.792z+0.8291)}{z(z-1)}. \tag{121b}$$

The non dominant pole of $T_d(z,\hat{\eta})$ is updated according to the identification of $\eta$ (REMES et al., 2021a; SILVA; CAMPESTRINI; BAZANELLA, 2016).

The $\mathcal{H}_\infty$ norm of $S(z,\rho)$ can be estimated according to the method presented in Section 6.1, with the acquired data set $\{u(k), y(k), k = 1...N\}$ and the VRFT-designed controller $C(z,\hat{\rho})$, resulting in $\hat{M}_S = 2.1952$, which may be too high for applications that require higher robustness indexes (SKOGESTAD; POSTLETHWAITE, 2005). The next subsection presents the application of the proposed method to reduce $M_S$ for the obtained VRFT solution.

### 7.1.3 Step 2 - Swarm intelligence algorithm

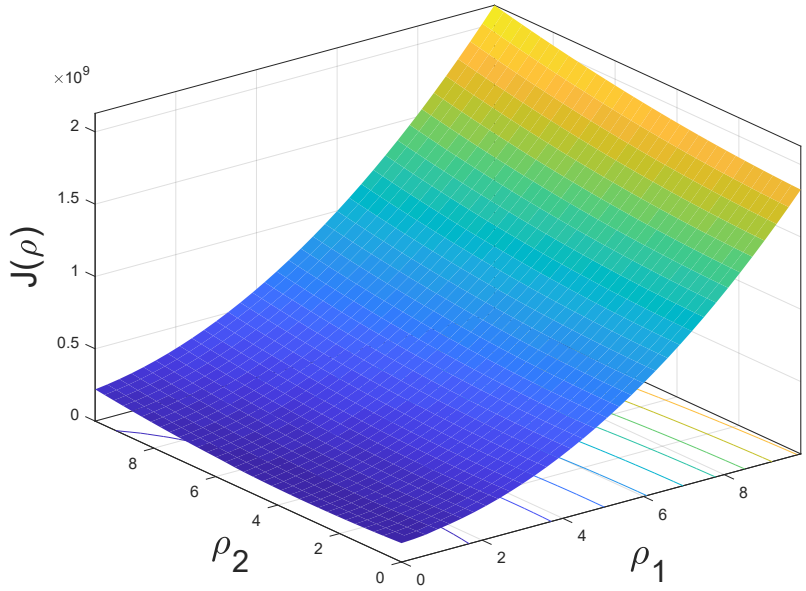
The four swarm intelligence algorithms presented in Chapter 5 - PSO, ABC, GWO, and I-GWO - were used to minimize the proposed cost function (110) considering an initial solution - or initial spawn central point - equal to the VRFT-designed controller in Subsection 7.1.2, $\rho_0 = [1.1246 \;\; 0.3124 \;\; 6.9713]'$.

The upper search bound is defined as $u_b = 10$, which should be sufficient considering that the maximum $\rho$ value of the VRFT-obtained controller is 6.9713 and, taking into account a choice of $M_{Sd}$ that is not too ambitious, the parameters values should not differ that much from the initial ones. The lower search bound, on the other hand, is set to $l_b = 0$ to avoid negative controller gain, which increases controller's passivity (BAO; PETER, 2007), and therefore, to ensure some degree of robustness. In order to better illustrate the search space, Figure 11a shows the search space in 3 dimensions considering parameters $\rho_1, \rho_2$, and Figure 11b exhibits the search space for the parameters $\rho_2, \rho_3$, where $\rho = [\rho_1 \;\; \rho_2 \;\; \rho_3]'$ and $J(\rho)$ is the cost.

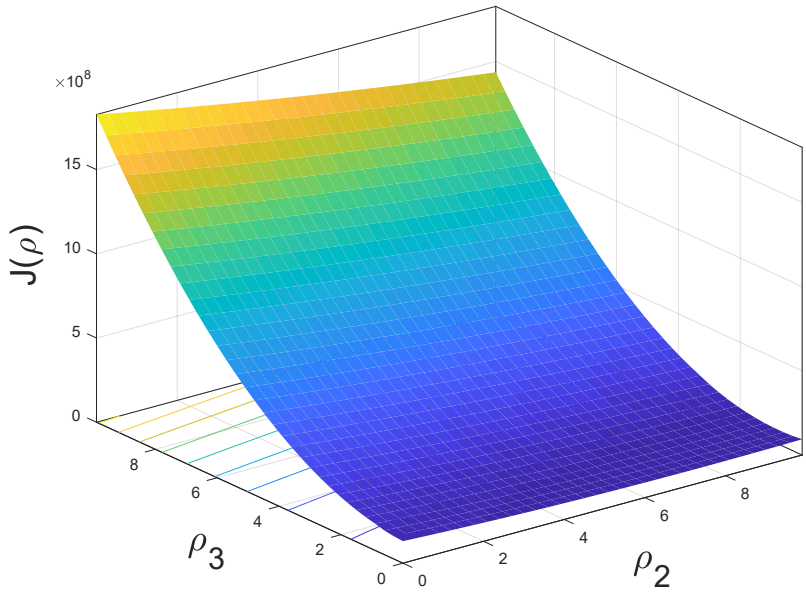
The reference model considered in the swarm intelligence optimization is $T_d(z,\hat{\eta})$. An initial population spawn radius of $R = (|u_b| + |l_b|)/2 = 5$ is used, as suggested in (113). The chosen desired $\mathcal{H}_\infty$ norm of $S(z,\rho)$ to be achieved is 1.8, which is sufficiently robust considering $M_{Sd} < 2$, as mentioned in the last chapters, and at the same time is not too low to result in a too poor performance. The constant $c$ to apply the penalty as shown in (111) is set arbitrarily as 1000, that satisfies $c >> 1$ (LUENBERGER; YE, 2015).

The number of agents of all algorithms is set as 50, as done in the basis literature for the PSO algorithm (KENNEDY; EBERHART, 1995), and the maximum number of iterations is set to 100. Each algorithms is run 50 times, with different realizations of the initial positions of the agents, with spawn radius as aforementioned. Such procedure can be characterized as a Monte Carlo experiment, since only one random variable (spawn) is being changed at each run. The hyperparameters for PSO and ABC, besides number of agents and iterations, are provided in

Figure 11 – Surface of the search space for example 1.




(a) $\rho_1, \rho_2$




(b) $\rho_2, \rho_3$

Source: the author.

Table 6, being chosen based on the Global Optimization Toolbox (THE MATHWORKS INC., 2021) for PSO and in an ABC algorithm implementation commonly mentioned in the literature (HERIS, 2015). For GWO and I-GWO, the only hyperparameters are the number of agents and maximum number of iterations.

At each iteration of the swarm intelligence algorithm, $\hat{M}_S$ is estimated with the technique proposed in Section 6.1, where the length of the identified impulse response is chosen, for each case, automatically by comparing the variation of two subsequent identified coefficients. If the variation is less than $1 \times 10^{-9}$, the algorithm stops and the IR is considered to be estimated.

Table 6 – Parameters settings for PSO and ABC.

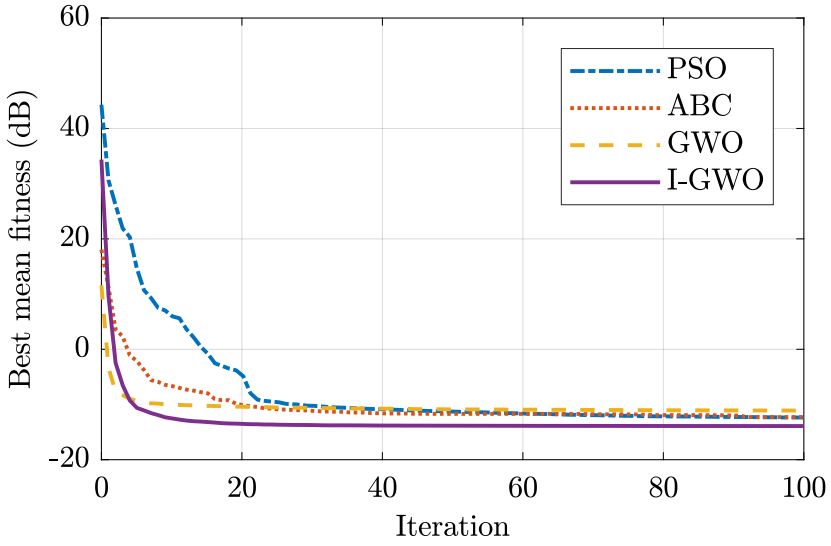
| Algorithm | Parameter settings | Value |
|---|---|---|
| PSO | Cognitive learning factor ($C_1$) | 1.49 |
| | Social learning factor ($C_2$) | 1.49 |
| | Inertia range (range of $w_1$) | [0.1,1.1] |
| ABC | Limit of food sources ($L$) | 90 |
| | Acceleration coefficient ($a$) | 1 |

Source: the author.

#### *7.1.3.1 Swarm intelligence algorithm results*

Once the hyperparameters are all set, the algorithms have been executed. The average convergence curves of all swarm intelligence algorithms for example 1 are shown in Figure 12. Table 7 shows the time that each algorithm took by iteration, considering the average of all 50 runs, and the number of iterations needed to converge, considering a $\delta = 1 \times 10^{-3}$ as criterion for defining convergence. The results were obtained in a personal computer with an Intel Core I5 4670 3.40 GHz processor, with 8 GB of RAM - DDR3 1600 MHz. The I-GWO algorithm took a longer time to converge, followed by ABC, PSO, and at last, GWO.

Figure 12 – Average convergence curves for all algorithms considering a Monte Carlo experiment of 50 runs for example 1.




Source: the author.

The box plot presented in Figure 13 regards the final fitness value obtained for all algorithms throughout the 50 runs. The most desired performance in terms of fitness, standard

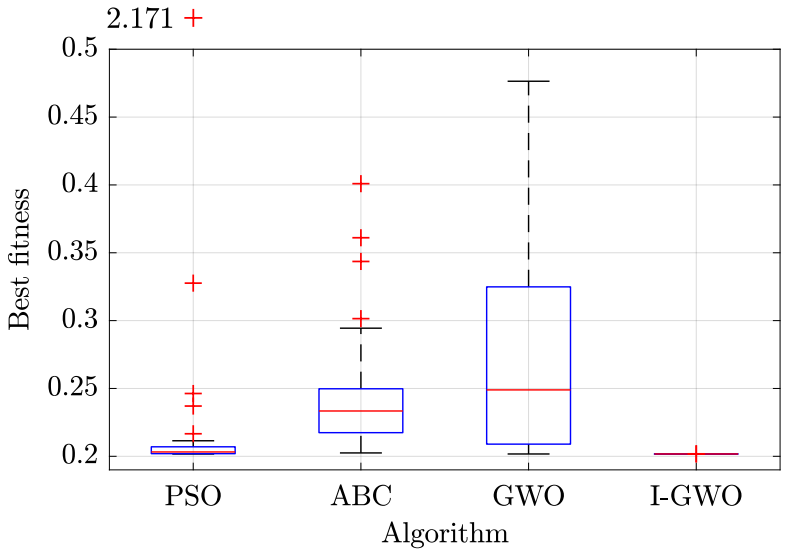
Table 7 – Time for convergence of all algorithms for example 1.

| Algorithm | 1-iteration time (s) | Iterations to converge | Time to converge (s) |
|---|---|---|---|
| PSO | 8.04 | 41 | 329.75 |
| ABC | 22.63 | 19 | 430.02 |
| GWO | 11.25 | 14 | 157.53 |
| I-GWO | 24.18 | 20 | 483.66 |

Source: the author.

deviation, and number of outliers is observed for I-GWO. In general, PSO, ABC, and GWO resulted in higher fitness than I-GWO for this example. Some of the relevant quantitative values regarding Figure 13 are presented in Table 8, which confirm the conclusions taken from the box plot.

Figure 13 – Box plot of a Monte Carlo experiment with 50 runs for all algorithms in terms of best fitness value obtained for example 1.




Source: the author.

Table 8 – Quantitative results from the box plot in terms of best fitness for example 1.

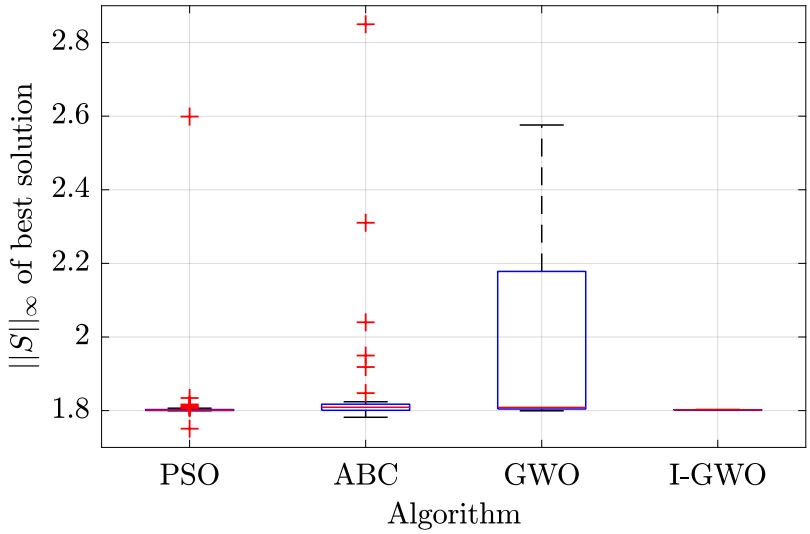
| Algorithm | median | $\sigma$ | min | max |
|---|---|---|---|---|
| PSO | 0.2032 | 0.2782 | 0.2017 | 2.1710 |
| ABC | 0.2334 | 0.0402 | 0.2025 | 0.4009 |
| GWO | 0.2489 | 0.0858 | 0.2017 | 0.4764 |
| I-GWO | 0.2017 | $5.0516 \times 10^{-5}$ | 0.2017 | 0.2018 |

Source: the author.

The box plot in Figure 14 presents the achieved values for the $\mathcal{H}_\infty$ norm of $S(z,\rho)$, by applying the the best solution (controller) obtained at each run of the algorithms in a closed-loop to $G_1(z)$, acquiring data with a PRBS signal with the same characteristics as aforementioned and estimating the $||S(z,\rho)||_\infty$ norms. The most desired result in terms of $\hat{M}_S$ considering the average value obtained, standard deviation, and number of outliers is achieved by I-GWO. PSO

obtained satisfactory results except from one outlier. ABC obtained three solutions that resulted in an $\hat{M}_S > 2$, whilst GWO did not obtain satisfactory results, since there are too many $\hat{M}_S$ values greater than 2. Notice that any solution with an $\hat{M}_S > 2.1952$ have a worse robustness than the VRFT-designed solution. The quantitative data from the box plot at Figure 14 is presented in Table 9, confirming the conclusions over the box plot. Similar conclusions regarding each swarm intelligence algorithm applied to the problem can be drawn from Figure 15, in which the parameter $\rho = [k_p \; k_i \; k_d]'$ identified for each algorithm is shown in the form of box plots.

Figure 14 – Box plot of 50 runs for all algorithms in terms of $||S||_\infty$ value obtained for example 1.




Source: the author.

Table 9 – Quantitative results from the box plot in terms of $||S||_\infty$ for example 1.

| Algorithm | median | $\sigma$ | min | max |
|---|---|---|---|---|
| PSO | 1.8014 | 0.1130 | 1.7509 | 2.5988 |
| ABC | 1.8091 | 0.4740 | $9.9798 \times 10^{-5}$ | 2.8496 |
| GWO | 1.8087 | 0.2923 | 1.7995 | 2.5760 |
| I-GWO | 1.8020 | $4.5561 \times 10^{-4}$ | 1.8008 | 1.8030 |

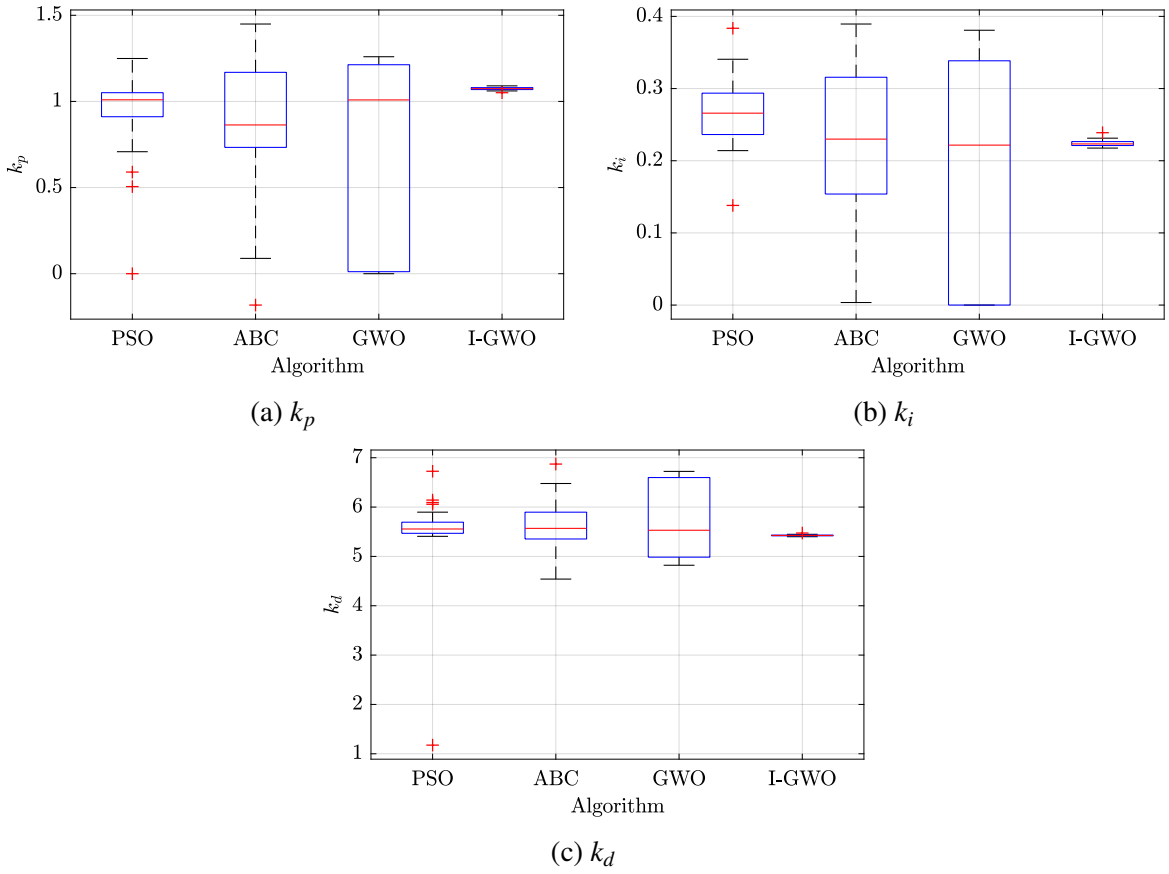
Source: the author.

To better illustrate the difference between the obtained controller with the swarm intelligence algorithms and the VRFT-obtained controller, one of the I-GWO solutions

$$\rho_{IGWO} = [1.090 \quad 0.2194 \quad 5.4018] \tag{122}$$

is applied to the system in the form of a controller $C_{IGWO} = \rho'_{IGWO}\bar{C}(z)$, as well as the initial VRFT controller in (121). For a step signal applied to the system in closed-loop as reference, the obtained output for the system considering both controllers is shown in Figure 17, with the controller signals shown in Figure 16. It can be clearly seen that the I-GWO-designed controller presents a smoother response, with less oscillations and reduced overshoot and undershoot, as

Figure 15 – Box plot of the obtained $k_p$ ($\rho(1)$), $k_i$ ($\rho(2)$), and $k_d$ ($\rho(3)$) controller parameters for all algorithms at 50 Monte Carlo experiments for example 1.




(a) $k_p$ (b) $k_i$

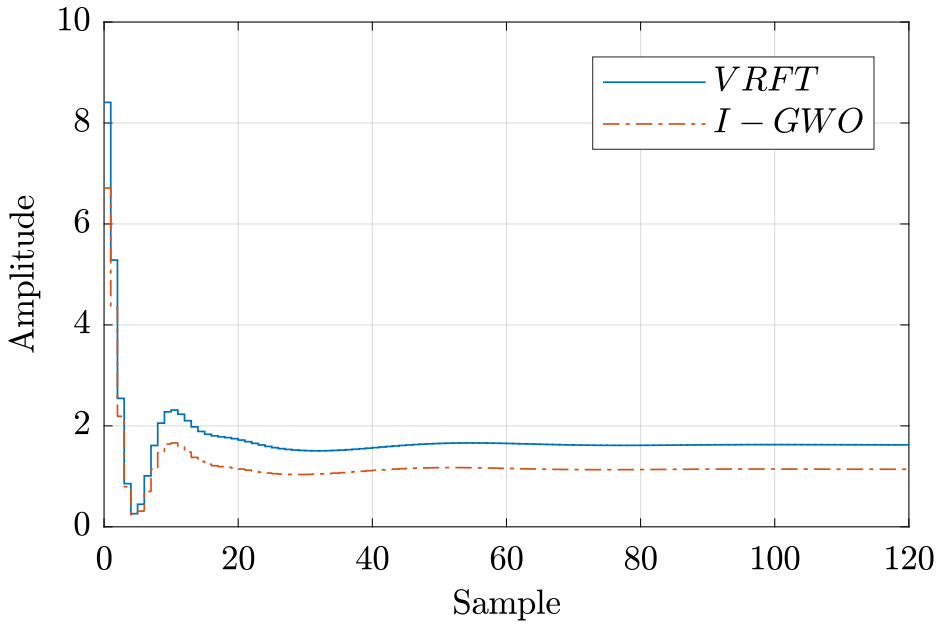
(c) $k_d$

Source: the author.

well as lower settling time. The observed controller effort for I-GWO is reduced if compared to the VRFT-designed controller. Obviously, the chosen controller class was not able to achieve the desired control requirements, even with the presence of a mismatch filter, presenting non-null overshoot and higher settling time than expected. This type of situation is where the proposed method of this thesis is better used, since it compensates for the demanding (or bad) choice of $T_d(z)$. Relevant information regarding the closed-loop system with the tested controllers in Figure 17 are presented in Table 10, where it is shown that the I-GWO-obtained controller achieved a lower overshoot and undershoot, with a $\hat{M}_S$ of 1.8030 - which reduction resulted also in higher gain and phase margins. The $M_S$ was estimated with a *validation* [1] PRBS signal with $N = 10000$ samples.

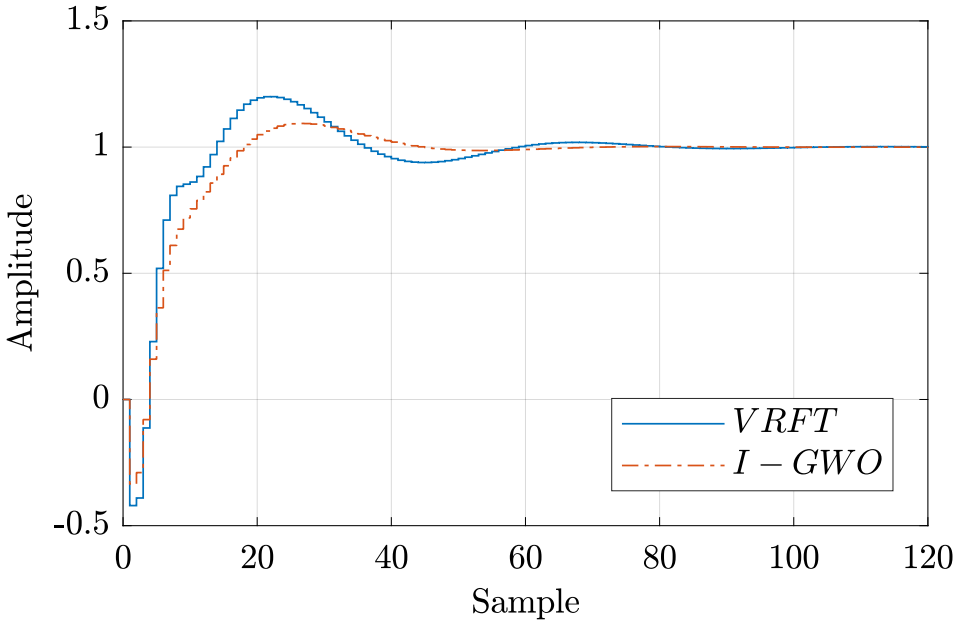
[1] Notice that the term *validation* is used here to represent a signal that is different from that used in the design.

Figure 16 – Control signal for a reference step signal applied to the controlled plant with a controller designed through the proposed method using I-GWO and the VRFT-designed controller for example 1.




Source: the author.

Figure 17 – Output signal for a reference step signal applied to the controlled plant with a controller designed through the proposed method using I-GWO and the VRFT-designed controller for example 1.




Source: the author.

Table 10 – Comparison between important characteristics of the closed-loop system with the I-GWO-designed and the VRFT-designed controller for example 1.

| Characteristic | VRFT | I-GWO |
|---|---|---|
| Settling time | 54 s | 39 s |
| Step overshoot | 20 % | 9 % |
| Step undershoot | 42 % | 33 % |
| $\hat{M}_S$ | 2.2079 | 1.8030 |
| Gain margin | 5.18 dB | 7 dB |
| Phase margin | 43 ° | 55.8 ° |

Source: the author.

## 7.2 EXAMPLE 2: FOURTH-ORDER PLANT

The process considered in example 2 consists of

$$G_2(z) = \frac{0.1381(z-0.95)(z^2-1.62z+0.6586)}{(z^2-1.7z+0.7325)(z^2-1.84z+0.8564)}, \tag{123}$$

which is similar to the discrete-time model of a SEPIC converter for the transfer function from duty cycle to output voltage (KASSICK, 2011) - a dc-dc converter which is, essentially, a Boost converter followed by a Buck-Boost. Since there is no NMP zeros in the plant, the VRFT method is used without the need of a flexible reference model criterion (BAZANELLA; CAMPESTRINI; ECKHARD, 2012). Notice that the presence of a NMP zero can be evaluated the same way as in Section 7.1.

### 7.2.1 Data collection

The data for example 2 is acquired following the same procedure as described in Section 7.1, with a PRBS signal of $N = 2000$ samples applied to the system, in a closed with a stabilizing controller, which is obtained as

$$k_p = \frac{0.5}{||G_2(z)||_\infty} = 0.3828. \tag{124}$$

To represent measurement noise, additive white Gaussian noise with an SNR of 20 dB is included in the system's output and fed back. The acquired input-output data set is $\{u(k), y(k), k = 1...N\}$.

### 7.2.2 Step 1 - VRFT

With the acquired data, a controller is designed through the VRFT as presented in Chapter 4. For example 2, the considered control requirements for the VRFT design are:

1. Null error in steady state for a step reference;
2. Settling time of approximately 6.5 times faster than the settling time in closed-loop with the stabilizing controller $k_p$;
3. Null overshoot for a step reference.

Notice that the settling time requirement is very ambitious and is representing, in this example, a poor choice of control requirements". The criteria for choosing the reference model $T_d(z)$ considering the aforementioned control requirements follows the same as depicted in Section 7.1. Notice that as the SEPIC is a dc-dc converter, the frequency of interest is kept as $\Omega_0 = 0$. Also, the poles are chosen in order to meet the settling time requirement, being kept real to avoid overshoot. Nevertheless, the reference model is

$$T_d(z) = \frac{1.4(z-0.6)}{(z-0.3)(z-0.2)}. \tag{125}$$

Suppose a situation where the controller is applied to the plant through limited hardware, where only a PI controller can be used. Thus, the controller class for this example is

$$\bar{C}(z) = \begin{bmatrix} 1 & \frac{z}{z-1} \end{bmatrix}'. \tag{126}$$

The obtained controller through the VRFT is given by the parameters

$$\rho = [6.6568 \quad 3.3728], \tag{127}$$

which, from (55), results in the controller

$$C(\rho, z) = \rho' \bar{C}(z) = \frac{10.03(z-0.6637)}{(z-1)}. \tag{128}$$

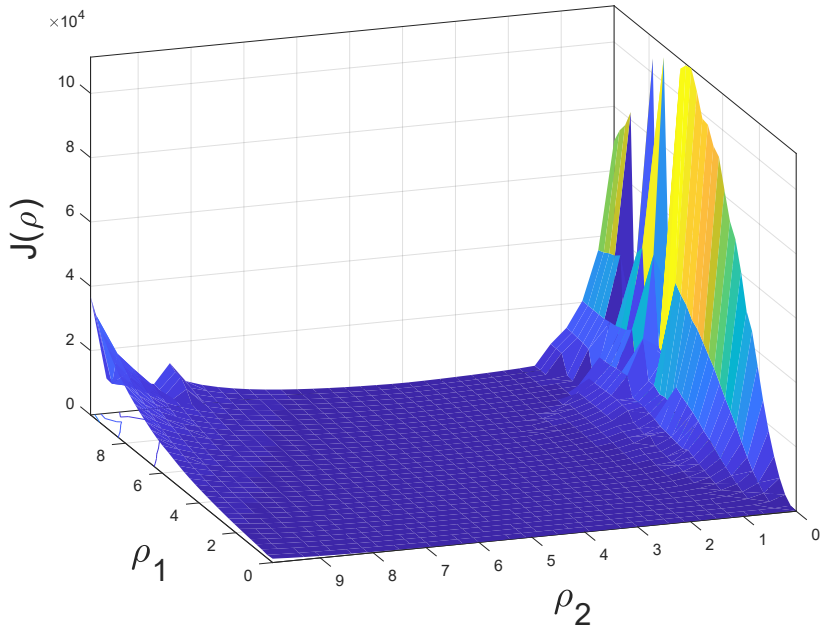
Estimating the $\mathcal{H}_\infty$ norm of $S(z,\rho)$ with the technique presented in Chapter 3 and the extension from Section 6.1, with data-set $\{u_0(k), y_0(k), k = 1...N\}$ and controller $C(z,\rho)$, an $\hat{M}_S = 2.2767$ is obtained. In order for the system to be sufficiently robust for general applications, an $M_S \leq 2$ is desired. Therefore, the proposed solution with swarm intelligence algorithms is applied.

### 7.2.3 Step 2 - Swarm intelligence algorithm

The swarm intelligence algorithms mentioned in Chapter 5 - PSO, ABC, GWO, and I-GWO - are considered to be applied to minimize the problem (110). The upper and lower search bounds are, respectively, $u_b = 10$ and $l_b = 0$, where the lower bound is chosen to increase the passivity of the controller (BAO; PETER, 2007). An upper bound of 10 should be sufficient, considering that the maximum desired robustness is not too far from the estimated robustness index at the end of step 1. In order to better illustrate the problem, Figure 18 presents the surface of the search space for this example, where $\rho = [\rho_1 \ \rho_2]'$ are the parameters of the PI controller and $J(\rho_1, \rho_2)$ is the cost, evaluated at each value of $\rho$. The agents of the optimization algorithms are spawned in a radius of $R = 5$ as suggested in (113). The desired $\mathcal{H}_\infty$ norm of $S(z)$ to be obtained, $M_{Sd}$, is set to 1.5. The penalty constant $c$ from (111) is set as 1000, which should be sufficient since $c >> 1$ (LUENBERGER; YE, 2015).

For all algorithms, 50 agents and 100 maximum iterations are set to the optimization procedure. Each algorithm is executed 50 times with different agent spawn realization at each run, but following the spawn radius suggestion, guaranteeing sufficient statistics to analyze the results, which characterizes a Monte Carlo experiment. The parameters of PSO and ABC are set as in Table 6. GWO and I-GWO algorithms do not require any additional hyperparameters to be set except from number of agents and maximum number of iterations. At each iteration, the IR is estimated with automatic choice of the number of coefficients, as aforementioned in the first example.

Figure 18 – Surface of the search space for example 2.




Source: the author.

#### *7.2.3.1 Swarm intelligence algorithm results*

For example 2, the average convergence curves of all algorithms are presented in Figure 19. The time that each algorithm took by iteration, considering the average of all 50 runs, and the number of iterations needed to converge, considering a $\delta = 1 \times 10^{-3}$ as criterion for defining convergence, is shown in Table 11. The results were obtained in a personal computer with an Intel Core I5 4670 3.40 GHz processor, with 8 GB of RAM - DDR3 1600 MHz. In this case, the ABC algorithm took the longest time to converge, followed by I-GWO, PSO, and GWO.

Table 11 – Time for convergence of all algorithms for example 2.

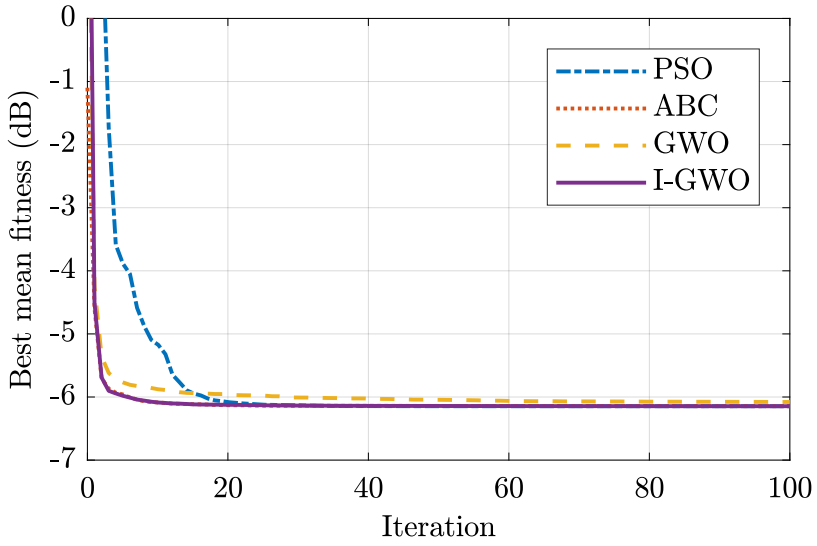
| Algorithm | 1-iteration time (s) | Iterations to converge | Time to converge (s) |
|---|---|---|---|
| PSO | 7.24 | 20 | 144.72 |
| ABC | 22.93 | 10 | 229.32 |
| GWO | 7.55 | 9 | 68.00 |
| I-GWO | 22.34 | 9 | 201.10 |

Source: the author.

The box plot with the best fitness values obtained for each algorithm is presented in Figure 20, where PSO, ABC, and I-GWO obtained a satisfactory behavior in terms of mean, standard deviation, and number of outliers, whilst GWO had more outliers than those aforementioned. The lower standard deviation is observed for the PSO algorithm. The conclusions over Figure 20 are in agreement with the quantitative values shown in Table 12.

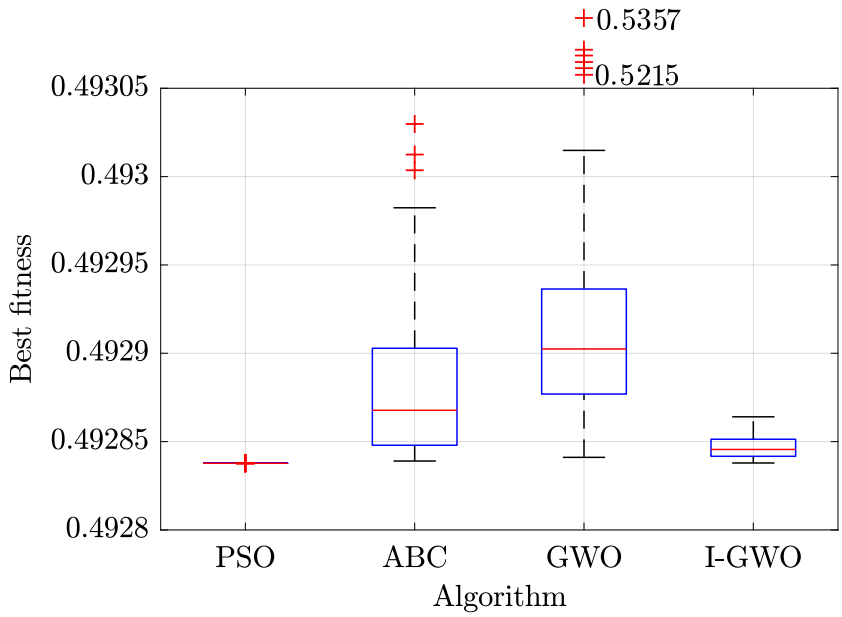
The $\mathcal{H}_\infty$ norm of $S(z,\rho)$, obtained with a *validation* PRBS signal of $N = 10000$ samples, considers the best solution for each algorithms obtained at each run and is presented in the form

Figure 19 – Average convergence curves for all algorithms considering a Monte Carlo experiment of 50 runs for example 2.




Source: the author.

Figure 20 – Box plot of a Monte Carlo experiment with 50 runs for all algorithms in terms of best fitness value obtained for example 2.




Source: the author.

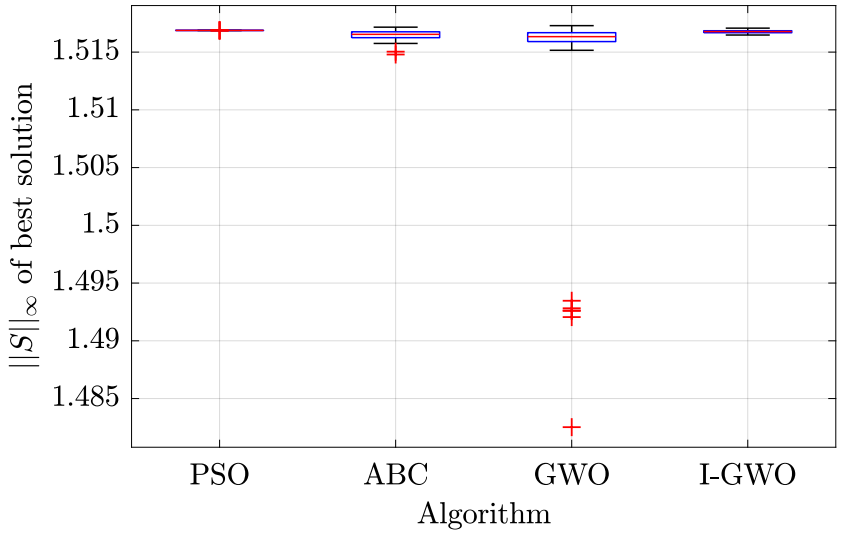
Table 12 – Quantitative results from the box plot in terms of best fitness for example 2.

| Algorithm | median | $\sigma$ | min | max |
|---|---|---|---|---|
| PSO | 0.49284 | $2.2270 \times 10^{-9}$ | 0.49284 | 0.49284 |
| ABC | 0.49287 | $4.7174 \times 10^{-5}$ | 0.49284 | 0.49303 |
| GWO | 0.49290 | $1.0414 \times 10^{-2}$ | 0.49284 | 0.53568 |
| I-GWO | 0.49285 | $6.4636 \times 10^{-6}$ | 0.49284 | 0.49286 |

Source: the author.

of a box plot in Figure 21, where its quantitative values are shown in Table 13. All algorithms presented similar median, except for GWO that had further outliers, satisfactory achieving close values to the desired one ($M_{Sd} \approx 1.5$). The I-GWO was the algorithm that had the least number of outliers, whilst the PSO had the lower standard deviation. Similar conclusions over the results can be also drawn from Figure 22, where the box plot of the obtained controller parameters $\rho = [k_p \; k_i]'$ are shown.

Figure 21 – Box plot of a Monte Carlo experiment with 50 runs for all algorithms in terms of $||S||_\infty$ for example 2.




Source: the author.

Table 13 – Quantitative results from the box plot in terms of $||S||_\infty$ for example 2.

| Algorithm | median | $\sigma$ | min | max |
|---|---|---|---|---|
| PSO | 1.5169 | $1.4558 \times 10^{-6}$ | 1.5169 | 1.5169 |
| ABC | 1.5165 | $4.4060 \times 10^{-4}$ | 1.5148 | 1.5172 |
| GWO | 1.5133 | $8.4447 \times 10^{-3}$ | 1.4825 | 1.5173 |
| I-GWO | 1.5168 | $1.2216 \times 10^{-4}$ | 1.5165 | 1.5171 |

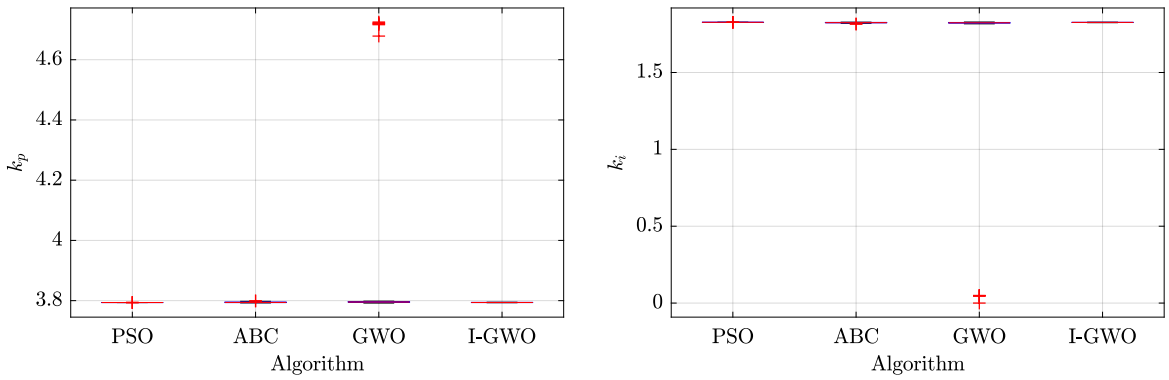
Source: the author.

To more clearly illustrate the difference between the controller obtained with swarm intelligence algorithms and the controller obtained via VRFT, lets consider one of the I-GWO solutions:

$$\rho_{IGWO} = [3.7946 \quad 1.8241]. \tag{129}$$

The I-GWO-obtained controller is applied to the system in simulation, as well as the VRFT controller from (127). A step signal is applied to both closed-loop systems and their output is presented in Figure 24, and their related controller signals is presented at Figure 23. Some relevant characteristics of the system are shown in Table 14, where the $\hat{M}_S$ value is obtained via data, following the method in Chapter 3, with a PRBS of $N = 10000$ samples as a *validation* signal. Differently from example 1, the proposed method did not reduce the settling time, which

Figure 22 – Box plot of the obtained $k_p$ ($\rho(1)$), $k_i$ ($\rho(2)$), and $k_d$ ($\rho(3)$) controller parameters for all algorithms at 50 Monte Carlo experiments for example 2.

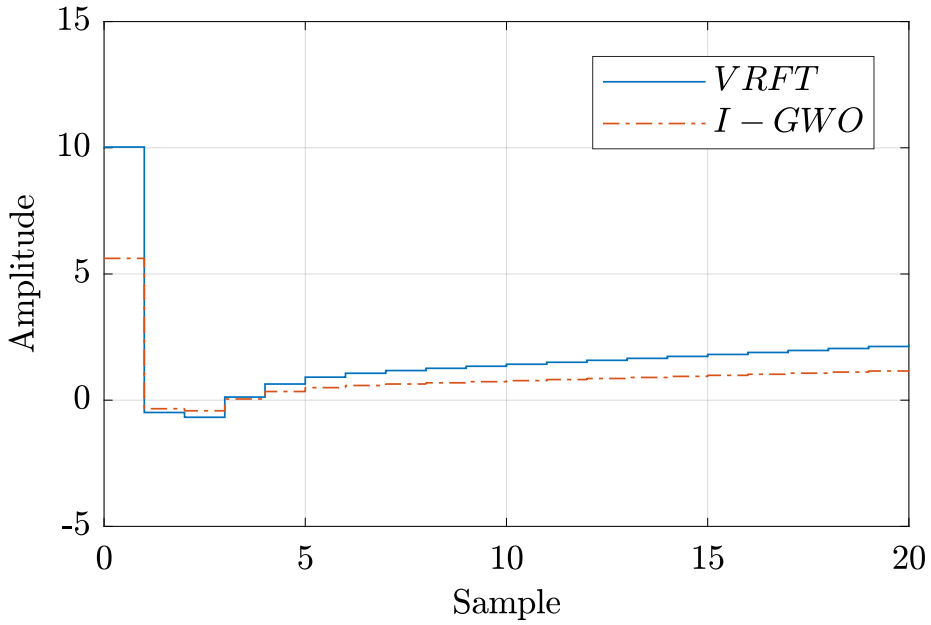



(a) $k_p$ (b) $k_i$

Source: the author.

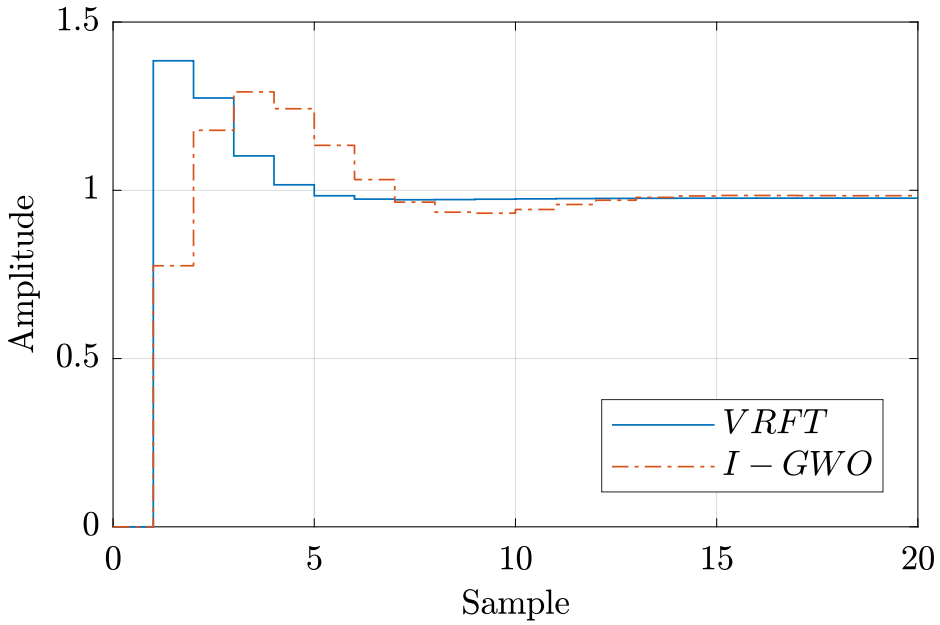
is expected since the penalty term will compensate the performance ($\mathcal{H}_2$, VRFT cost function) part. The overshoot was reduced from 42 % to 29 %. Since the $M_S$ value was reduced to 1.5165, the gain and phase margins were increased altogether. Notice that, in a real scenario, trespassing the gain or phase margin values may result in, e.g., higher overshoot or even instability, resulting in a fault, which is not admissible for several systems, such as power converters. Therefore, the "sacrifice" of settling time is justified in cases where higher robustness is needed.

Figure 23 – Control signal for a reference step signal applied to the controlled plant with a controller designed through the proposed method using I-GWO and the VRFT-designed controller for example 2.




Source: the author.

Figure 24 – Output signal for a reference step signal applied to the controlled plant with a controller designed through the proposed method using I-GWO and the VRFT-designed controller for example 1.




Source: the author.

Table 14 – Comparison between important characteristics of the closed-loop system with the I-GWO-designed and the VRFT-designed controller for example 2.

| Characteristic | VRFT | I-GWO |
|---|---|---|
| Settling time | 5 s | 14 s |
| Step overshoot | 42 % | 29 % |
| $\hat{M}_S$ | 2.3834 | 1.5165 |
| Gain margin | 4.72 dB | 9.70 dB |
| Phase margin | 37.8 ° | 43.1 ° |

Source: the author.

## 7.3 FINAL CONSIDERATIONS

In this chapter, the application of the proposed technique was considered for two examples - a second-order plant based on a Boost converter and a fourth-order plant based on the structure of a SEPIC converter. For both cases, the assumed design requirements were too “aggressive” and resulted in an $M_S > 2$ for the controller designed with the VRFT. Nevertheless, the swarm intelligence algorithms were applied to reduce the $M_S$ to 1.8 for the first example, and 1.5 for the second example. For example 1, only the I-GWO obtained satisfactory results. In the case of example 2, except from GWO, all algorithms achieved satisfactory results, with PSO and I-GWO presenting the most desired results. For example 1, the slowest algorithm to converge was I-GWO, and the fastest was GWO. On the other hand, example 2 showed the slowest convergence behavior for ABC, whilst GWO was still the fastest of all algorithms to reach convergence.

# 8 CONCLUSION

This thesis proposed a robust solution to the Virtual Reference Feedback Tuning design method by including a robustness constraint, in the form of a penalty, in its cost function, which is solved via swarm intelligence algorithms and still maintains the necessity of only a single batch of data of the VRFT. The proposed method was addressed and summarized in Chapter 6.

The $\mathcal{H}_\infty$ norm of the sensitivity function is estimated using data as proposed in Chapter 3, with the extension to a single-batch of data as shown in Chapter 6, at each iteration of the swarm intelligence algorithm, via impulse response of the system, which is obtained in a regularized fashion through the empirical Bayes method. The proposed method for estimating system norms have been tested through Monte Carlo experiments and compared to a state-of-the-art technique, presenting satisfactory results and lowering the mean percent error if compared to the literature-available method for SISO systems.

The optimization procedure done with swarm intelligence algorithms aims to achieve a desired value of $\mathcal{H}_\infty$ norm of the sensitivity transfer function ($M_{Sd}$) while maintaining the lowest possible cost in terms of reference tracking - the VRFT cost function, $\mathcal{H}_2$-based - by changing the controller parameters. Four swarm intelligence algorithms - PSO, ABC, GWO, and I-GWO - are used to minimize the problem and illustrate the proposed method with two plants inspired on the structure of a Boost and a SEPIC converter in Chapter 7. For each illustrative example, an initial controller is designed through VRFT resulting in an $M_S > 2$. Then, the swarm intelligence algorithms are applied to both cases, with the objective of reducing the $\mathcal{H}_\infty$ norm of the sensitivity function from a value greater than 2 to the desired norm value.

For each algorithm, 50 runs are executed with different realization of agents' initialization. For example 1 (plant $G_1$), faster convergence was achieved by GWO, whilst the slowest was observed for I-GWO. I-GWO was the algorithm that presented the least number of outliers and the lower standard deviation regarding both best fitness evaluated at the cost function and the obtained $M_S$ value. The median value of estimated $M_S$ for all algorithms was close to the desired constraint value, as expected. For example 2 (plant $G_2$), on the other hand, ABC showed the slowest convergence time, wilst GWO was still the fastest of all algorithms to converge. PSO, again, was the slowest algorithm to converge. Regarding best fitness value, PSO and I-GWO presented the least number of outliers and lower standard deviation, while ABC presented a still satisfactory behavior, and GWO achieved some further (and undesired) outliers. PSO obtained the lower standard deviation in this case. Taking a look at the estimated $\mathcal{H}_\infty$ norm of the sensitivity transfer function for the second example, all solutions were acceptable, with median close to the desired bound. Except from GWO, the other algorithms did not present noticeable outliers.

The main contributions of this thesis can be summarized according to the following topics:

1. The proposal of a swarm intelligence-based robust solution for VRFT;

2. A data-driven one-shot method for estimating system norms purely based on impulse response, with its extension to a single-batch of data for the proposed problem;

3. Development and release of a Python package for non-parametric data-driven impulse response estimation (see Appendix A), resulting in the publication: "*impulseest*: A Python package for non-parametric impulse response estimation with input–output data" (FIORIO; REMES; NOVAES, 2021).

As for future works, it is suggested:

1. The inclusion of other constraints (e.g., for control effort and performance) simultaneously with the robustness constraint in the optimization problem (109), to achieve specific performance behavior whilst increasing the overall robustness;

2. Comparison of the proposed method with other type of metaheuristic optimization algorithms, as evolutionary or physics-based;

3. Obtention of an approximate convex relation to the problem (109), allowing for the use of simpler optimization techniques;

4. Application of a robustness constraint to the OCI, VDFT, or DD-LQR techniques;

5. Extend the current work for MIMO systems.

# BIBLIOGRAPHY

AGUIAR, Renato A. et al. Fractional pid controller applied to a chemical plant with level and ph control. **Chemical Product and Process Modeling**, v. 13, 2018. Cited in page 21.

ALATAS, Bilal; CAN, Umit. Physics based metaheuristic optimization algorithms for global optimization. **American Journal of Information Science and Computer Engineering**, 01 2015. Cited in page 22.

ALCÁNTARA, S.; VILANOVA, R.; PEDRET, C. Pid control in terms of robustness/performance and servo/regulator trade-offs: A unifying approach to balanced autotuning. **Journal of Process Control**, v. 23, n. 4, p. 527–542, 2013. ISSN 0959-1524. Cited in page 21.

APKARIAN, Pierre; NOLL, Dominikus. Optimization-based control design techniques and tools. In: ____. **Encyclopedia of Systems and Control**. London: Springer London, 2019. p. 1–11. ISBN 978-1-4471-5102-9. Disponível em: <https://doi.org/10.1007/978-1-4471-5102-9_144-2>. Cited in page 24.

ÅSTRÖM, Karl J. Book. **Introduction to stochastic control theory**. [S.l.]: Academic Press New York, 1970. xi, 299 p. p. Cited in page 25.

ATHANS, M. The role and use of the stochastic linear-quadratic-gaussian problem in control system design. **IEEE Transactions on Automatic Control**, v. 16, n. 6, p. 529–552, 1971. Cited in page 25.

ATHANS, Michael. A tutorial on the lqg/ltr method. In: **1986 American Control Conference**. [S.l.: s.n.], 1986. p. 1289–1296. Cited in page 25.

BAO, Jie; PETER, Lee. **Process Control: The Passive Systems Approach**. [S.l.]: Springer Publishing Company, Incorporated, 2007. ISBN 978-1-84628-892-0. Cited 2 times in pages 69 and 78.

BATTISTELLI, Giorgio; SELVI, Daniela; TESI, Alberto. Robust switching control: Stability analysis and application to active disturbance attenuation. **IEEE Transactions on Automatic Control**, v. 62, n. 12, p. 6369–6376, 2017. Cited in page 26.

BAZANELLA, Alexandre Sanfelice; CAMPESTRINI, Lucíola; ECKHARD, Diego. **Data-Driven Controller Design: The H2 Approach**. [S.l.]: Springer Publishing Company, Incorporated, 2012. ISBN 940178356X. Cited 7 times in pages 21, 22, 47, 48, 50, 51, and 77.

BERBERICH, Julian et al. Data-driven model predictive control with stability and robustness guarantees. **IEEE Transactions on Automatic Control**, v. 66, n. 4, p. 1702–1717, 2021. Cited in page 28.

BHUSHAN, Ravi; CHATTERJEE, Kalyan; SHANKAR, Ravi. Comparison between ga-based lqr and conventional lqr control method of dfig wind energy system. In: **2016 3rd International Conference on Recent Advances in Information Technology (RAIT)**. [S.l.: s.n.], 2016. p. 214–219. Cited in page 25.

BOEIRA, E. C.; ECKHARD, D. Multivariable virtual reference feedback tuning with bayesian regularization. In: SBA SOCIEDADE BRASILEIRA DE AUTOMÁTICA. **XXII Congresso Brasileiro de Automática**. João Pessoa: SBA Sociedade Brasileira de Automática, 2018. p. 1–8. N/a. Cited in page 38.

BOYD, S. et al. **Linear Matrix Inequalities in System and Control Theory**. Philadelphia, PA: SIAM, 1994. v. 15. (Studies in Applied Mathematics, v. 15). ISBN 0-89871-334-X. Cited in page 25.

BRUNTON, S. L.; KUTZ, J. N. **Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control**. 1st. ed. USA: Cambridge University Press, 2019. ISBN 1108422098. Cited in page 67.

CAMPESTRINI, L. et al. Virtual reference feedback tuning for non-minimum phase plants. **Automatica**, v. 47, n. 8, p. 1778–1784, 2011. ISSN 0005-1098. Cited 6 times in pages 21, 47, 52, 63, 66, and 67.

CAMPESTRINI, Lucíola et al. Data-driven model reference control design by prediction error identification. **Journal of the Franklin Institute**, v. 354, n. 6, p. 2628–2647, 2017. ISSN 0016-0032. Special issue on recent advances on control and diagnosis via process measurements. Cited 3 times in pages 21, 26, and 27.

CAMPI, M.C.; LECCHINI, A.; SAVARESI, S.M. Virtual reference feedback tuning: a direct method for the design of feedback controllers. **Automatica**, v. 38, n. 8, p. 1337–1346, 2002. ISSN 0005-1098. Cited 5 times in pages 21, 26, 47, 49, and 50.

CARLIN, B. P.; LOUIS, T. A. Bayes and empirical bayes methods for data analysis. **Statistics and Computing**, Kluwer Academic Publishers, USA, v. 7, n. 2, p. 153–154, jun. 1997. ISSN 0960-3174. Cited 2 times in pages 30 and 38.

CHAUDHURI, Nilanjan Ray; CHAKRABORTY, Debraj; CHAUDHURI, Balarko. Damping control in power systems under constrained communication bandwidth: A predictor corrector strategy. **IEEE Transactions on Control Systems Technology**, v. 20, n. 1, p. 223–231, 2012. Cited in page 21.

CHEN, T.; LJUNG, L. Implementation of algorithms for tuning parameters in regularized least squares problems in system identification. **Automatica**, v. 49, n. 7, p. 2213–2220, 2013. ISSN 0005-1098. Cited 2 times in pages 93 and 94.

CHEN, T.; OHLSSON, H.; LJUNG, L. On the estimation of transfer functions, regularizations and gaussian processes—revisited. **Automatica**, v. 48, n. 8, p. 1525–1535, 2012. ISSN 0005-1098. Cited 7 times in pages 30, 31, 36, 37, 38, 43, and 94.

CHILUKA, S.K. et al. A novel robust virtual reference feedback tuning approach for minimum and non-minimum phase systems. **ISA Transactions**, v. 115, p. 163–191, 2021. Cited 3 times in pages 22, 27, and 35.

COLANERI, Patrizio; GEROMEL, José C.; ASTOLFI, Alessandro. Stabilization of continuous-time switched nonlinear systems. **Systems & Control Letters**, v. 57, n. 1, p. 95–103, 2008. ISSN 0167-6911. Cited in page 26.

COMMUNITY, The SciPy. **SciPy Optimization**. [S.l.], 2020. Disponível em: <https://docs.scipy.org/doc/scipy/reference/tutorial/optimize.html>. Cited 2 times in pages 38 and 95.

DEAECTO, G.S. et al. Switched affine systems control design with appliction to dc-dc converters. **Control Theory & Applications, IET**, v. 4, p. 1201 – 1210, 08 2010. Cited in page 26.

DORIGO, Marco; BIRATTARI, Mauro; STUTZLE, Thomas. Ant colony optimization. **IEEE Computational Intelligence Magazine**, v. 1, n. 4, p. 28–39, 2006. Cited in page 34.

DOYLE, J. Guaranteed margins for lqg regulators. **IEEE Transactions on Automatic Control**, v. 23, n. 4, p. 756–757, 1978. Cited in page 25.

DOYLE, J.C. et al. State-space solutions to standard h2 and hinf control problems. **IEEE Transactions on Automatic Control**, v. 34, n. 8, p. 831–847, 1989. Cited in page 25.

DU, Ke-Lin; SWAMY, M. N. S. **Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature**. 1st. ed. [S.l.]: Birkhäuser Basel, 2016. ISBN 3319411918. Cited 4 times in pages 32, 33, 54, and 55.

DUPONT, Fabricio Hoff et al. Comparison of linear quadratic controllers with stability analysis for dc-dc boost converters under large load range. **Asian Journal of Control**, v. 15, p. 861–871, 2013. Cited in page 25.

ECKHARD, D.; CAMPESTRINI, L.; BOEIRA, E. C. Virtual disturbance feedback tuning. **IFAC Journal of Systems and Control**, v. 3, p. 23–29, 2018. ISSN 2468–6018. N/a. Cited 3 times in pages 21, 26, and 27.

EGIDIO, Lucas N.; DEAECTO, Grace S. Novel practical stability conditions for discrete-time switched affine systems. **IEEE Transactions on Automatic Control**, v. 64, n. 11, p. 4705–4710, 2019. Cited in page 26.

ERICKSON, Robert W.; MAKSIMOVIC, Dragan. **Fundamentals of Power Electronics**. 2ed. ed. [S.l.]: Springer, 2001. Cited 2 times in pages 49 and 66.

FEO, Thomas; RESENDE, Mauricio. Greedy randomized adaptive search procedures. **Journal of Global Optimization**, v. 6, p. 109–133, 03 1995. Cited in page 33.

FIORIO, L. V. **impulseest**. [S.l.]: GitHub, 2021. <https://github.com/LVF784/impulseest>. Cited 2 times in pages 95 and 101.

FIORIO, L. V.; REMES, C. L.; NOVAES, Y. R. de. impulseest: A python package for non-parametric impulse response estimation with input–output data. **SoftwareX**, v. 15, p. 100761, 2021. ISSN 2352-7110. Cited 11 times in pages 38, 46, 67, 85, 93, 96, 97, 98, 99, 100, and 101.

GANDOMI, Amir Hossein; ALAVI, Amir Hossein. Krill herd: A new bio-inspired optimization algorithm. **Communications in Nonlinear Science and Numerical Simulation**, v. 17, n. 12, p. 4831–4845, 2012. ISSN 1007-5704. Cited in page 34.

GLOVER, Fred; LAGUNA, Manuel. Tabu search. In: ____. **Handbook of Combinatorial Optimization: Volume1–3**. Boston, MA: Springer US, 1998. p. 2093–2229. ISBN 978-1-4613-0303-9. Cited in page 33.

GOLUB, G. H.; LOAN, C. F. Van. **Matrix Computations (3rd Ed.)**. USA: Johns Hopkins University Press, 1996. ISBN 0801854148. Cited 2 times in pages 94 and 95.

GREEN, Michael; LIMEBEER, David J. N. **Linear Robust Control**. USA: Prentice-Hall, Inc., 1994. ISBN 0131022784. Cited 2 times in pages 24 and 25.

HEIDARI, Ali Asghar; PAHLAVANI, Parham. An efficient modified grey wolf optimizer with lévy flight for optimization tasks. **Applied Soft Computing**, v. 60, p. 115–134, 2017. ISSN 1568-4946. Cited in page 58.

HERIS, M. K. **The Yarpiz Project**. [S.l.], 2015. Disponível em: <https://yarpiz.com/about>. Cited in page 70.

HJALMARSSON, Håkan. Iterative feedback tuning. **IFAC Proceedings Volumes**, v. 31, n. 22, p. 101–108, 1998. ISSN 1474-6670. IFAC Workshop on Adaptive Systems in Control and Signal Processing 1998, Glasgow, Scotland, 26-28 August 1998. Cited 2 times in pages 21 and 26.

HUANG, H. et al. Uncertainty optimization of pure electric vehicle interior tire/road noise comfort based on data-driven. **Mechanical Systems and Signal Processing**, v. 165, 2022. Cited in page 21.

JOHNSON, R.; LUCAS, G. **STAR WARS: EPISODE VIII - THE LAST JEDI**. USA: Walt Disney Studios Motion Pictures, 2018. Produced by Lucasfilme. Cited in page 5.

JONES, A. et al. **Lateralus**. [S.l.]: Volcano Entertainment, 2002. Cited in page 6.

KAMMER, Leonardo C.; BITMEAD, Robert R.; BARTLETT, Peter L. Direct iterative tuning via spectral analysis. **Automatica**, v. 36, n. 9, p. 1301–1307, 2000. ISSN 0005-1098. Cited in page 26.

KARABOGA, Dervis; BASTURK, Bahriye. Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems. In: MELIN, Patricia et al. (Ed.). **Foundations of Fuzzy Logic and Soft Computing**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 789–798. ISBN 978-3-540-72950-1. Cited 4 times in pages 22, 34, 54, and 55.

KARIMI, Alireza; KAMMER, Christoph. A data-driven approach to robust control of multivariable systems by convex optimization. **Automatica**, v. 85, p. 227–233, 2017. ISSN 0005-1098. Cited 3 times in pages 27, 28, and 35.

KARIMI, A.; MIŠKOVIĆ, L.; BONVIN, D. Iterative correlation-based controller tuning. **International Journal of Adaptive Control and Signal Processing**, v. 18, n. 8, p. 645–664, 2004. Cited 3 times in pages 21, 26, and 27.

KASSICK, Enio V. Deriving the canonical equivalent circuit for small signal & low frequency ac model for the sepic and zeta pwm dc-dc converters with two-port network (quadripoles) circuit analysis technique. **Eletr. Potên.**, v. 16, p. 376–382, November 2011. Cited in page 77.

KAZIMIERCZUK, M.K. **Pulse-width Modulated DC-DC Power Converters**. [S.l.]: Wiley, 2008. ISBN 9780470694657. Cited 2 times in pages 21 and 49.

KEEL, L. H.; BHATTACHARYYA, S. P. Controller synthesis free of analytical models: Three term controllers. **IEEE Transactions on Automatic Control**, v. 53, n. 6, p. 1353–1369, 2008. Cited in page 21.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: **Proceedings of ICNN'95 - International Conference on Neural Networks**. [S.l.: s.n.], 1995. v. 4, p. 1942–1948 vol.4. Cited 4 times in pages 22, 34, 54, and 69.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **Science**, v. 220, n. 4598, p. 671–680, 1983. Cited in page 33.

KOBAKU, Tarakanath; PATWARDHAN, Sachin C.; AGARWAL, Vivek. Experimental evaluation of internal model control scheme on a dc–dc boost converter exhibiting nonminimum phase behavior. **IEEE Transactions on Power Electronics**, v. 32, n. 11, p. 8880–8891, 2017. Cited in page 21.

KUČERA, Vladimir. A method to teach the parameterization of all stabilizing controllers. **IFAC Proceedings Volumes**, v. 44, n. 1, p. 6355–6360, 2011. ISSN 1474-6670. 18th IFAC World Congress. Cited in page 25.

LEUNG, F.H.F.; TAM, P.K.S.; LI, C.K. The control of switching dc-dc converters-a general lwr problem. **IEEE Transactions on Industrial Electronics**, v. 38, n. 1, p. 65–71, 1991. Cited in page 25.

LJUNG, Lennart. **System Identification (2nd Ed.): Theory for the User**. USA: Prentice Hall PTR, 1999. ISBN 0136566952. Cited 4 times in pages 32, 43, 50, and 63.

LONG, Wen et al. An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. **Engineering Applications of Artificial Intelligence**, v. 68, 04 2018. Cited in page 58.

LU, Chao; GAO, Liang; YI, Jin. Grey wolf optimizer with cellular topological structure. **Expert Systems with Applications**, v. 107, p. 89–114, 2018. ISSN 0957-4174. Cited in page 58.

LUENBERGER, David G.; YE, Yinyu. **Linear and Nonlinear Programming**. [S.l.]: Springer Publishing Company, Incorporated, 2015. ISBN 3319188410. Cited 5 times in pages 22, 30, 31, 69, and 78.

LUKE, Sean. **Essentials of Metaheuristics**. second. [S.l.]: Lulu, 2013. Cited in page 32.

MIRJALILI, Seyedali. **Genetic Algorithm**. Cham: Springer International Publishing, 2019. 43–55 p. ISBN 978-3-319-93025-1. Cited in page 22.

MIRJALILI, Seyedali; LEWIS, Andrew. The whale optimization algorithm. **Advances in Engineering Software**, v. 95, p. 51–67, 2016. ISSN 0965-9978. Cited in page 34.

MIRJALILI, Seyedali; MIRJALILI, Seyed Mohammad; LEWIS, Andrew. Grey wolf optimizer. **Advances in Engineering Software**, v. 69, p. 46–61, 2014. ISSN 0965-9978. Cited 5 times in pages 22, 33, 34, 54, and 57.

MÜLLER, M. I.; ROJAS, C. R. Iterative $\mathcal{H}_\infty$-norm estimation using cyclic-prefixed signals. In: **2020 59th IEEE Conference on Decision and Control (CDC)**. [S.l.: s.n.], 2020. p. 2869–2874. Cited in page 29.

NA, Jing et al. Output-feedback robust control of uncertain systems via online data-driven learning. **IEEE Transactions on Neural Networks and Learning Systems**, v. 32, n. 6, p. 2650–2662, 2021. Cited in page 28.

NADIMI-SHAHRAKI, Mohammad H.; TAGHIAN, Shokooh; MIRJALILI, Seyedali. An improved grey wolf optimizer for solving engineering problems. **Expert Systems with Applications**, v. 166, p. 113917, 2021. ISSN 0957-4174. Cited 4 times in pages 22, 34, 54, and 58.

NICOLETTI, Achille; MARTINO, Michele; KARIMI, Alireza. A robust data-driven controller design methodology with applications to particle accelerator power converters. **IEEE Transactions on Control Systems Technology**, v. 27, n. 2, p. 814–821, 2019. Cited 3 times in pages 27, 28, and 35.

NISE, Norman S. **Control Systems Engineering**. 3rd. ed. USA: John Wiley & Sons, Inc., 2000. ISBN 0471366013. Cited 4 times in pages 21, 43, 45, and 96.

OOMEN, T. et al. Iterative data-driven $\mathcal{H}_\infty$ norm estimation of multivariable systems with application to robust active vibration isolation. **IEEE Transactions on Control Systems Technology**, v. 22, n. 6, p. 2247–2260, 2014. Cited in page 30.

PÉREZ, Julio Ariel Romero; LLOPIS, Roberto Sanchis. Tuning and robustness analysis of event-based pid controllers under different event-generation strategies. **International Journal of Control**, Taylor & Francis, v. 91, n. 7, p. 1567–1587, 2018. Cited in page 21.

REMES, C. L. et al. Virtual reference feedback tuning applied to dc–dc converters. **IEEE Transactions on Industrial Electronics**, v. 68, n. 1, p. 544–552, 2021. Cited 4 times in pages 21, 66, 68, and 69.

REMES, Chrystian L. et al. Lqg controller in cascade loop tuned by pso applied to a dc–dc converter. **Asian Journal of Control**, v. 23, n. 5, p. 2370–2380, 2021. Cited 2 times in pages 25 and 50.

SILVA, Gustavo R. G. da; BAZANELLA, Alexandre S.; CAMPESTRINI, Lucíola. One-shot data-driven controller certification. **ISA transactions**, v. 99, p. 361—373, April 2020. ISSN 0019-0578. Cited 4 times in pages 9, 30, 45, and 46.

SILVA, Gustavo R. G. da et al. Data-driven lqr control design. **IEEE Control Systems Letters**, v. 3, n. 1, p. 180–185, 2019. Cited 2 times in pages 26 and 27.

SILVA, Gustavo R. G. da; CAMPESTRINI, Lucíola; BAZANELLA, Alexandre S. Multivariable vrft: an approach for systems with non-minimum phase transmission zeros. In: **2016 IEEE Conference on Control Applications (CCA)**. [S.l.: s.n.], 2016. p. 1324–1329. Cited in page 69.

SKOGESTAD, Sigurd; POSTLETHWAITE, Ian. **Multivariable feedback control: Analysis and Design**. Hoboken, US-NJ: John Wiley, 2005. Cited 12 times in pages 22, 24, 25, 29, 36, 39, 40, 41, 49, 66, 69, and 96.

TACX, P.; OOMEN, T. Accurate $\mathcal{H}_\infty$-norm estimation via finite-frequency norms of local parametric models. In: **2021 American Control Conference (ACC)**. [S.l.: s.n.], 2021. p. 332–337. Cited in page 29.

TALBI, El-Ghazali. **Metaheuristics: From Design to Implementation**. [S.l.]: John Wiley, 2009. v. 74. ISBN 9780470278581. Cited 4 times in pages 22, 32, 33, and 54.

TAN, N. van et al. Position control for haptic device based on discrete-time proportional integral derivative controller. **International Journal of Electrical and Computer Engineering**, v. 12, n. 1, p. 269–276, 2022. Cited in page 21.

THARANIDHARAN, V. et al. Robust finite-time pid control for discrete-time large-scale interconnected uncertain system with discrete-delay. **Mathematics and Computers in Simulation**, v. 192, p. 370–383, 2022. Cited in page 21.

THE MATHWORKS INC. **System Identification Toolbox Release 2017b**. Natick, Massachussetts, United States, 2021. Cited 2 times in pages 67 and 70.

TOIVONEN, Hannu. **Lecture notes in Advanced Control Methods**. [S.l.]: Åbo Akademi University, 2010. Cited 2 times in pages 39 and 40.

TROFINO, A. et al. Switching rule design for switched dynamic systems with affine vector fields. **Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference**, p. 6365–6370, 2009. Cited in page 26.

TROFINO, Alexandre et al. Stabilizing switching rule design for affine switched systems. In: **2011 50th IEEE Conference on Decision and Control and European Control Conference**. [S.l.: s.n.], 2011. p. 1183–1188. Cited in page 26.

TROFINO, Alexandre et al. Switching rule design for affine switched systems with $\mathcal{H}_\infty$ performance. In: **2012 IEEE 51st IEEE Conference on Decision and Control (CDC)**. [S.l.: s.n.], 2012. p. 1923–1928. Cited in page 26.

TU, Qiang; CHEN, Xuechen; LIU, Xingcheng. Hierarchy strengthened grey wolf optimizer for numerical optimization and feature selection. **IEEE Access**, v. 7, p. 78012–78028, 2019. Cited in page 58.

TUDON-MARTINEZ, J.C. et al. Advanced temperature control applied on an industrial box furnace. **Journal of Thermal Science and Engineering Applications**, v. 14, n. 6, 2022. Cited in page 21.

WAHAB, Mohd Nadhir Ab; NEFTI-MEZIANI, Samia; ATYABI, Adham. A comprehensive review of swarm optimization algorithms. **PLOS ONE**, Public Library of Science, v. 10, 05 2015. Cited in page 22.

WOLPERT, D.H.; MACREADY, W.G. No free lunch theorems for optimization. **IEEE Transactions on Evolutionary Computation**, v. 1, n. 1, p. 67–82, 1997. Cited 3 times in pages 22, 34, and 54.

XIE, Ruichao; KAMWA, Innocent; CHUNG, C. Y. A novel wide-area control strategy for damping of critical frequency oscillations via modulation of active power injections. **IEEE Transactions on Power Systems**, v. 36, n. 1, p. 485–494, 2021. Cited in page 21.

YANG, Xin-She. Firefly algorithm, stochastic test functions and design optimisation. **International Journal of Bio-inspired Computation**, v. 2, 03 2010. Cited in page 34.

YERRAMILLI, S.; TANGIRALA, A. **sysid: system identification in R**. [S.l.], 2017. Cited in page 67.

ZENELIS, I.; WANG, X. A model-free sparse wide-area damping controller for inter-area oscillations. **International Journal of Electrical Power and Energy Systems**, v. 136, 2022. Cited in page 21.

ZHOU, K.; DOYLE, J.C. **Essentials of Robust Control**. [S.l.]: Prentice Hall, 1998. (Essentials of Robust Control). ISBN 9780137908745. Cited 2 times in pages 24 and 25.

# APPENDIX A – PYTHON IMPLEMENTATION OF THE *IMPULSEEST* FUNCTION

A Python package for the non-parametric estimation of impulse response using input-output data of a system was developed during the course of this thesis, resulting in the published paper (FIORIO; REMES; NOVAES, 2021). This appendix describes in details the implementation of the package and presents the main results of the paper.

## A.1 SOFTWARE ARCHITECTURE

The package has two main modules, *creation.py* and *impulseest.py*. The first module contains mainly initialization functions, to create necessary vectors and matrices that are called by *impulseest.py*. Those functions are:

- **create_alpha**: returns the initial $\alpha$ hyperparameter vector, according to the chosen regularization kernel;
- **create_bounds**: returns the upper and lower bounds for each hyperparameter in the vector generated by *create_alpha*;
- **create_Phi**: creates and returns the vector of input regressors;
- **create_Y**: creates and returns the vector of output regressors.

The objective of placing boundaries at the hyperparameters of $\alpha$ is to guarantee a low condition number for the covariance matrix $P_n$. The upper and lower bounds for each hyperparameter is suggested in (CHEN; LJUNG, 2013), except for the lower bound of $c$, which was set to $1 \times 10^{-8}$ in order to avoid computational problems that was occurring during the test phases of the package. All hyperparameters' bounds are shown in Table 15.

Table 15 – Hyperparameters' bounds.

| Parameter | Lower bound | Upper bound |
|---|---|---|
| c | $10^{-8}$ | none ($\infty$) |
| $\lambda$ | 0.7 (DI, TC) \| 0.72 (DC) | 1 |
| $\sigma$ | 0 | none ($\infty$) |
| $\iota$ (DC only) | -0.99 | 0.99 |

Source: (FIORIO; REMES; NOVAES, 2021).

The module *impulseest.py* is formed by two functions:

- **impulseest**: solves the cost function which results in the IR estimates;
- **argument_check**: checks if the arguments entered by the user are valid.

The *impulseest* function is the main function of the whole package and is described in detail at Subsection A.2.

## A.2 FUNCTIONALITIES

The *impulseest* is the function responsible for the non-parametric estimation of IR coefficients based on input-output data by minimizing cost function (21) in a computational efficient way (detailed further in this subsection). The function has five possible input arguments:

- **u** [numpy array]: input signal (size $N \times 1$);
- **y** [numpy array]: output signal (size $N \times 1$);
- **n** [integer]: number of impulse response estimates (default is $n = 100$);
- **RegularizationKernel** [string]: regularization kernel - 'none', 'DC', 'DI' or 'TC' (default is 'none');
- **MinimizationMethod** [string]: bound-constrained optimization method used to minimize the cost function - 'L-BFGS-B', 'Powell' or 'TNC' (default is 'L-BFGS-B').

The input array $u$ and output array $y$ are reshaped to an $N \times 1$ shape in case they are not provided that way. All input arguments are checked by *argument_check* before the impulse response estimation. If an error is detected, an exception is raised to the user. All initial matrices and arrays are, then, initialized by *creation.py* module.

There are two local functions inside *impulseest* function. The *Prior* function is responsible for creating the prior covariance matrix $P_n$ at each iteration, according to the current $\alpha$ values via expressions (28), depending on the chosen regularization kernel.

The *cost_function* is the function that implements (21) and calls *Prior* at each iteration. The regularized least squares can be implemented efficiently in two main ways: using the Cholesky factorization or the QR factorization (GOLUB; LOAN, 1996). If $P_n$ is ill-conditioned, QR factorization presents more precise results than Cholesky factorization (GOLUB; LOAN, 1996). Literature has shown that $P_n$ can be very ill-conditioned for IR estimation (CHEN; OHLSSON; LJUNG, 2012). Therefore, the QR factorization-based algorithm (CHEN; LJUNG, 2013) and is described below.

Firstly, the thin QR factorization (GOLUB; LOAN, 1996) is precomputed as

$$[\Phi_N^T \; Y_N] = Q_d[R_{d1} \; R_{d2}], \tag{130}$$

where $Q_d$ is an $N \times (n+1)$ matrix, $R_{d1}$ is an $(n+1) \times n$ matrix, and $R_{d2}$ is an $(n+1) \times 1$ array. After computing (130), *cost_function* is called and execute the following steps:

1. compute the Cholesky factorization $L$ of $P_n(\alpha)$ (GOLUB; LOAN, 1996);
2. compute $R_{d1}L$;

3. compute the QR factorization (GOLUB; LOAN, 1996)

$$\begin{bmatrix} R_{d1}L & R_{d2} \\ \sigma I_n & 0 \end{bmatrix} = Q_c R_c; \tag{131}$$

4. compute the cost

$$\frac{r^2}{\sigma^2} + (N-n)\log\sigma^2 + 2\log|R_1|. \tag{132}$$

The implementation in (FIORIO, 2021) uses Scipy's optimization package (COMMUNITY, 2020) to solve the cost function, allowing the user to choose between 'L-BFGS-B', 'Powell' or 'TNC' bound-constrained methods. Finally, the *impulseest* function returns the $n$ first impulse response coefficients.

## A.3 ILLUSTRATIVE EXAMPLES

The *impulseest* is illustrated through an experimental case and through a collection of transfer functions considering a whole grid of values - denoted here as test grid. Also, three Monte Carlo experiments are executed for each regularization kernel. Quantitative and qualitative analysis are made considering the obtaining results, mainly via box plots.

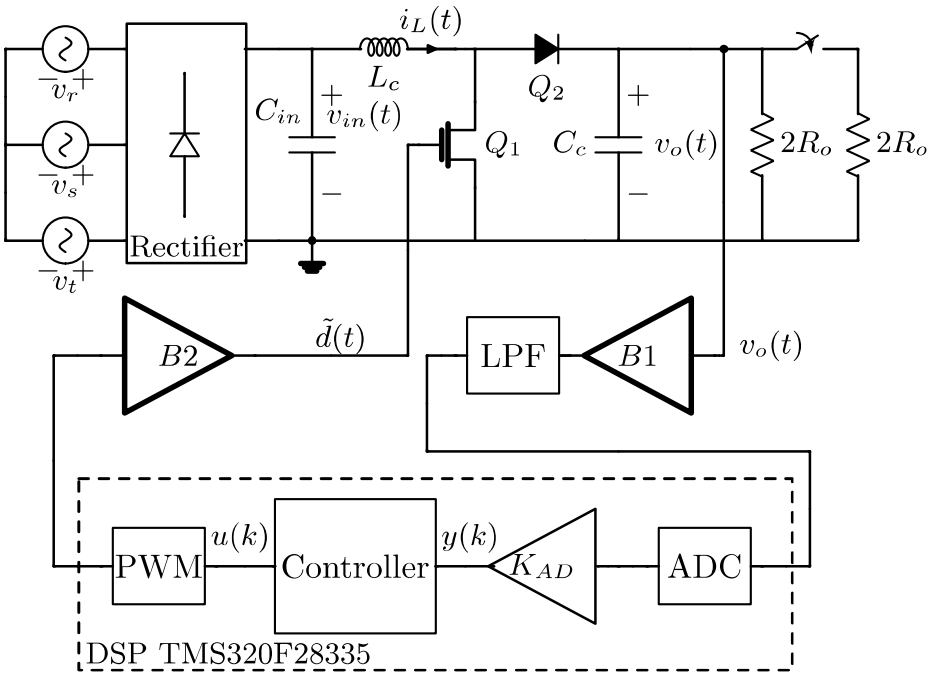
### A.3.1 Experimental case

The experimental case embraces a Boost converter operating in continuous conduction mode, designed to increase the voltage of a photovoltaic array. The converter is controlled by a Digital Signal Processor (DSP) with a stabilizing proportional controller, as per the control scheme shown in Figure 25. The parameters are presented in Table 16. The labels from Figure 25 that are not defined in Table 16 are: $v_r$, $v_s$, $v_t$ - phase input voltage of the rectifier; $i_L(t)$ - inductor current; $Q_1$ - MOSFET transistor; $Q_2$ - output diode; $v_o(t)$ - instantaneous output voltage; $R_o$ - output load resistance; B1 and B2 - DSP buffers; LPF - low-pass filter; ADC - analog-to-digital converter; $K_{AD}$ - feedback gain; $y(k)$ - digital output voltage signal; $u(k)$ - controller's output; PWM - pulse-width modulation block; and $\tilde{d}(t)$ - duty cycle signal, applied to $Q_1$.

The input-output signals for IR estimation are obtained as follows: an excitation signal with a square waveform with 37 Hz of frequency, generated by the DSP, is used as reference signal for $\tilde{d}(t)$ and applied to the gate of the switch $Q_1$; the output voltage signal $v_o(t)$ is acquired from the Boost converter. Both signals are stored in the DSP memory and saved into a personal computer and are presented in Figure 26.

The IR is then estimated using the *impulseest* package with no regularization and with regularization through the DC kernel (28), which result is shown in Figure 27. The non-regularized estimation present higher variance, as expected. At the other hand, the increasing variance is compensated with the DC kernel, resulting in a less noisy and more precise estimation.

Figure 25 – Rectifier, Boost converter and the control scheme for this example.




Source: (FIORIO; REMES; NOVAES, 2021).

Table 16 – Parameters of the boost converter used as example.

| Description | Parameter | Value |
|---|---|---|
| Output power | $P_o$ | 400 W |
| Input voltage | $v_{in}$ | 65 V $\sim$ 85 V |
| Nominal output voltage | $Y_o$ | 310 V |
| Nominal duty cycle | $U_o$ | 0.72 pu |
| Switching frequency | $f_s$ | 50 kHz |
| Sampling frequency | $f_a$ | 50 kHz |
| Output filter inductance | $L_c$ | 2.15 mH |
| Output filter capacitance | $C_c$ | 2.2 $\mu$F |
| Input capacitance | $C_{in}$ | 22.6 mF |
| Cut-off frequency LPF | $\omega_{LPF}$ | 25 kHz |

Source: (FIORIO; REMES; NOVAES, 2021).

### A.3.2 Test grid

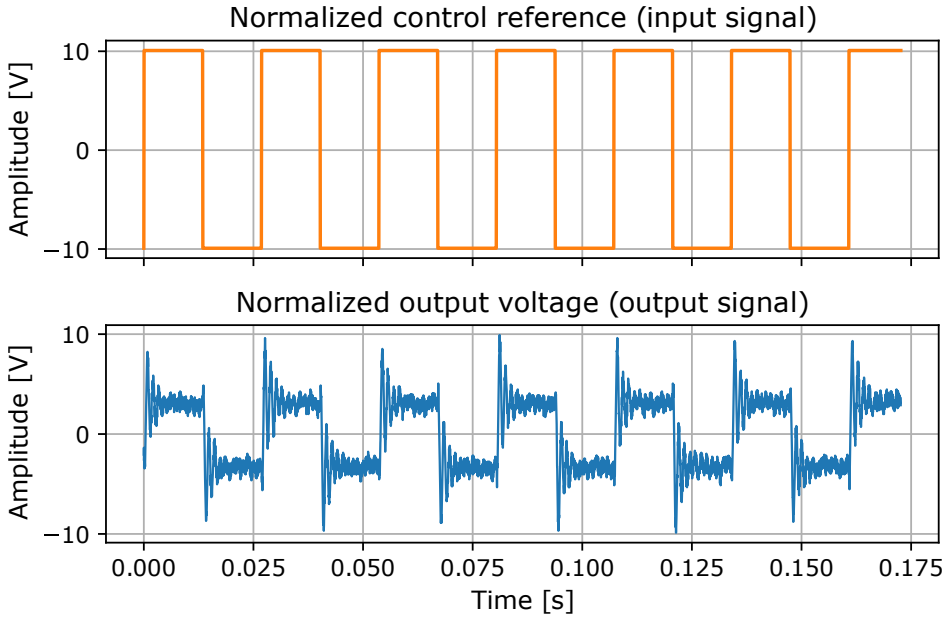
A second-order discrete-time transfer function was defined for the test grid, since many processes can be modeled as such (NISE, 2000; SKOGESTAD; POSTLETHWAITE, 2005). The considered transfer function has the structure of

$$G_{test}(z) = \frac{p_1 + p_2}{\lambda} \frac{(z-\lambda)}{(z-p_1)(z-p_2)}, \tag{133}$$

with a time-step of 0.001 seconds and parameters defined as follows:

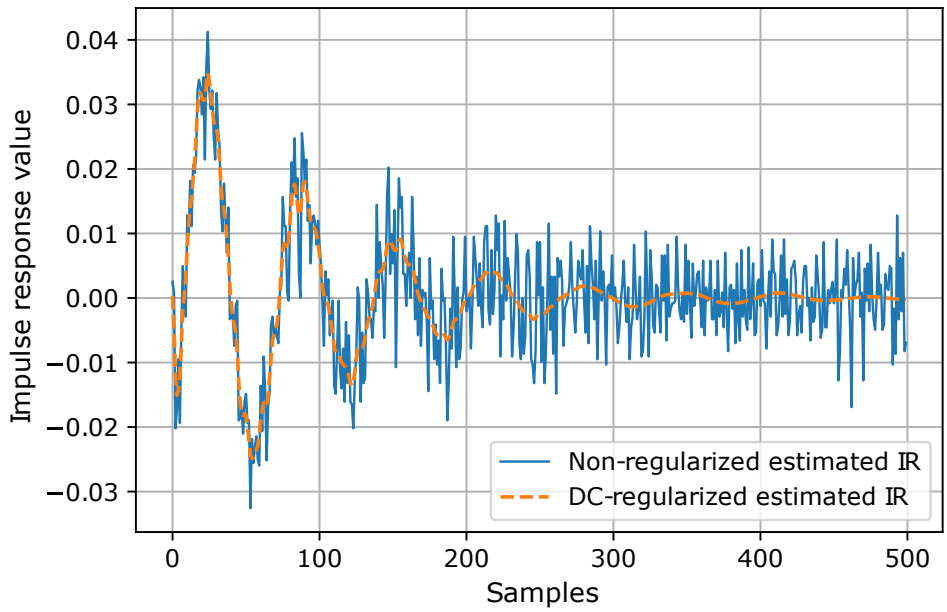
- the zero position $\lambda$ varies from 0.22 to 3.16, including both minimum and non-minimum phase zeros. The case where $\lambda = 0$ is not included for simplicity;

Figure 26 – Normalized control reference signal (input) and normalized output voltage (output) of the boost converter used for this example.




Source: (FIORIO; REMES; NOVAES, 2021).

Figure 27 – DC-regularized and non-regularized impulse response estimation of a boost converter's output voltage by control reference plant.




Source: (FIORIO; REMES; NOVAES, 2021).

- the poles are complex-conjugate poles, positioned according to $p_{1,2} = a\mathrm{e}^{\pm j \cdot b}$, with $a \in (0.27,\ 0.978)$ and $b \in (0.006,\ 0.82)$. All considered poles are within the unity circle, resulting in stable systems.

The values assumed by $a$, $b$, and $\lambda$ are presented in Table 17 and result in 512 possible combinations.

Table 17 – Values of $a$, $b$, and $\lambda$.

| a | b | $\lambda$ |
|---|---|---|
| 0.272251 | 0.006738 | 0.223130 |
| 0.473223 | 0.013376 | 0.325814 |
| 0.598901 | 0.026554 | 0.475753 |
| 0.697120 | 0.052715 | 0.694693 |
| 0.779969 | 0.104649 | 1.014388 |
| 0.852692 | 0.207748 | 1.481207 |
| 0.918117 | 0.412419 | 2.162854 |
| 0.977970 | 0.818731 | 3.158193 |

Source: (FIORIO; REMES; NOVAES, 2021).

A PRBS with time-step of 0.001 seconds and $N = 2000$ samples is used as excitation signal for the test grid. Each plant is simulated using SciPy's *dlsim* function, obtaining an output signal. Additive white Gaussian noise with an SNR of 6 dB was included in all data before the estimations. The results are compared in terms of MSE in relation to the model-based (real) calculated impulse response considering the first 200 terms of the IR.

Figure 28 shows a box plot with all MSEs obtained, for each kernel, considering all possible parameters combination according to Table 17. Each parameter combination was run twice, with different noise realizations, resulting in a total of 1024 runs for each kernel. Table 18 presents the quantitative values for the box plot in Figure 28. A lower MSE is observed at the regularized cases in comparison with the non-regularized case. The most desired behavior is seen at the TC and DC kernels. For each box plot, there are around 200 outliers out of the 1024 samples. The presence of outliers is expected since for each combination of parameters and realization of noise, a different type of regularization may achieve better results.

Table 18 – Quantitative box plot results for the grid test MSEs.

| Metrics | none | TC | DI | DC |
|---|---|---|---|---|
| mean ($\times 10^{-3}$) | 20.0 | 2.12 | 25.7 | 1.37 |
| $\sigma$ ($\times 10^{-2}$) | 26.3 | 3.53 | 40.4 | 1.98 |
| min ($\times 10^{-12}$) | 36.9 | 7.40 | 10.7 | 16.9 |
| max | 6.16 | 1.01 | 10.1 | 0.50 |

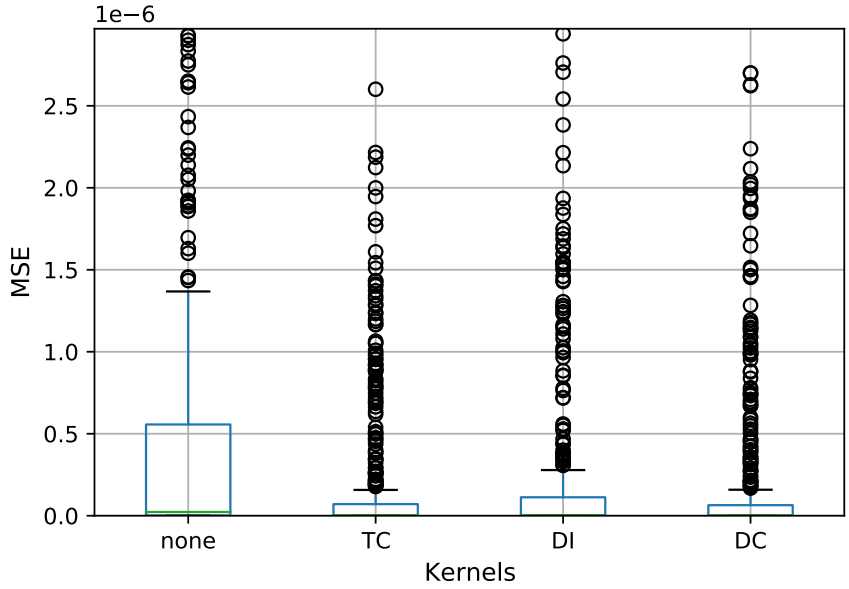
Source: (FIORIO; REMES; NOVAES, 2021).

### A.3.3 Monte Carlo experiments

In this sense, a Monte Carlo experiment is to estimate the IR of a discrete-time plant with fixed parameters, with different noise realization at each run. Based on structures that are common to appear in processes that can be approximated to second-order models, three plants have been considered:

$$G_1(z) = -0.25\frac{(z-1.2)}{(z-0.95)}; \tag{134}$$

Figure 28 – Box plot of all the MSEs obtained during the test for all kernels.




Source: (FIORIO; REMES; NOVAES, 2021).

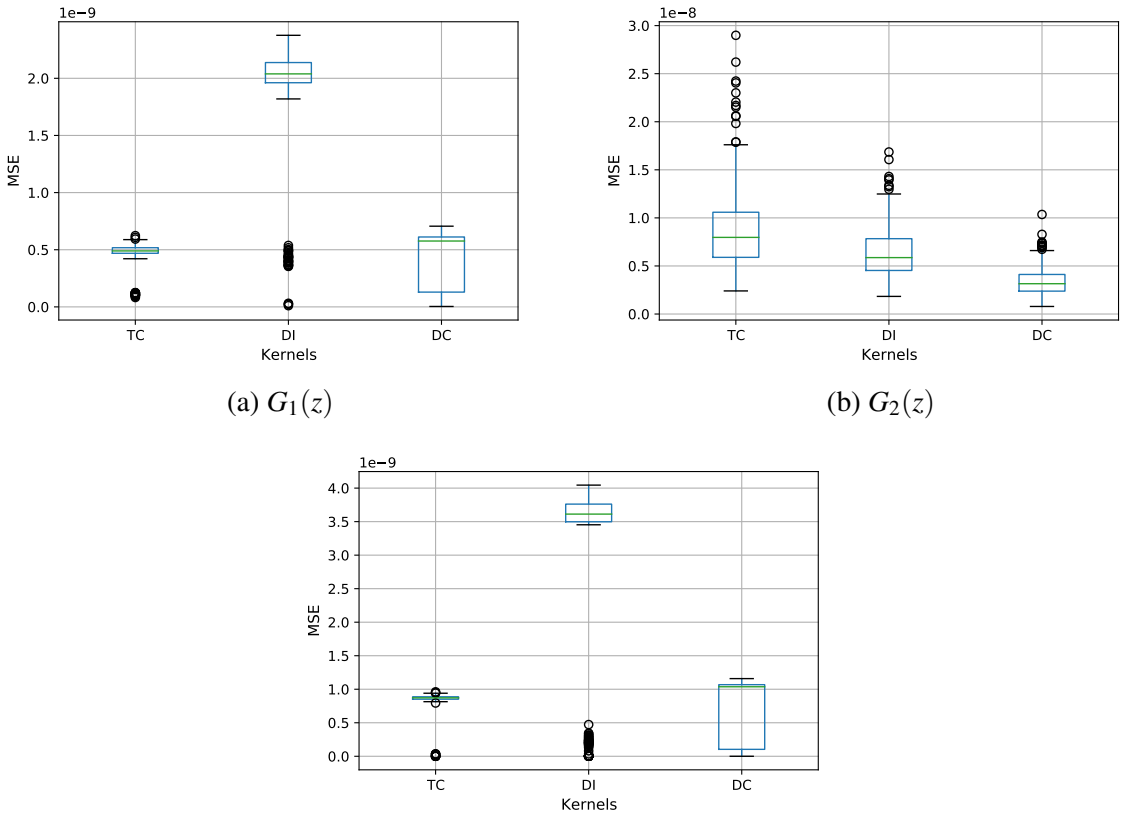
$$G_2(z) = 0.5\frac{(z-0.8)}{(z-0.9-0.3j)(z-0.9+0.3j)}; \tag{135}$$

$$G_3(z) = -0.075\frac{(z-1.2)}{(z-0.7)(z-0.95)}. \tag{136}$$

300 runs are considered for each regularization kernel. At each run, the data is corrupted with additive Gaussian noise with an SNR of 6 dB.

Figures 29a, 29b, and 29c present the box plot of the obtained MSE at the estimation of the IR of plants $G_1(z)$, $G_2(z)$, and $G_3(z)$, respectively, when compared to its model-based (real) calculated IR. For plant $G_1(z)$, the TC kernel obtained the smallest interquartile range with a few outliers. The DC kernel did not obtain outliers, although had the greatest interquartile range. DI kernel presented a higher median than the other two kernels, being the least desired behavior for this case. For plant $G_3(z)$, Figure 29c, the behavior of all three kernels were the same as those observed for plant $G_1(z)$ at the box plot in Figure 29a. At the other hand, considering plant $G_2(z)$, Figure 29b, the most desired behavior is clearly obtained by the DC kernel - smallest interquartile range, least number of outliers, and lower median. The DI and TC kernels presented, respectively, the second and third most desired behavior of all three kernels. The same conclusion can be obtained from the quantitative values regarding the box plots, shown in Tables 19, 20, and 21, for plants $G_1(z)$, $G_2(z)$, and $G_3(z)$, respectively.

Figure 29 – Box plot of the Monte Carlo MSEs for plants $G_1(z)$, $G_2(z)$ and $G_3(z)$.




(a) $G_1(z)$ (b) $G_2(z)$

(c) $G_3(z)$

Source: (FIORIO; REMES; NOVAES, 2021).

Table 19 – Quantitative box plot results (MSE) for plant $G_1(z)$.

| Metrics | TC | DI | DC |
|---|---|---|---|
| mean ($\times 10^{-10}$) | 4.73 | 18.7 | 4.44 |
| $\sigma$ ($\times 10^{-11}$) | 9.85 | 55.4 | 24.3 |
| min ($\times 10^{-12}$) | 82.1 | 10.6 | 3.64 |
| max ($\times 10^{-10}$) | 6.23 | 23.8 | 7.06 |

Source: (FIORIO; REMES; NOVAES, 2021).

Table 20 – Quantitative box plot results (MSE) for plant $G_2(z)$.

| Metrics | TC | DI | DC |
|---|---|---|---|
| mean ($\times 10^{-9}$) | 8.86 | 6.38 | 3.43 |
| std ($\times 10^{-9}$) | 4.32 | 2.50 | 1.44 |
| min ($\times 10^{-10}$) | 24.0 | 18.3 | 7.86 |
| max ($\times 10^{-8}$) | 2.90 | 1.69 | 1.04 |

Source: (FIORIO; REMES; NOVAES, 2021).

Table 21 – Quantitative box plot results (MSE) for plant $G_3(z)$.

| Metrics | TC | DI | DC |
|---|---|---|---|
| mean ($\times 10^{-10}$) | 8.44 | 30.4 | 7.52 |
| std ($\times 10^{-10}$) | 1.57 | 13.6 | 4.61 |
| min ($\times 10^{-13}$) | 3.44 | 13.3 | 1.62 |
| max ($\times 10^{-10}$) | 9.61 | 40.4 | 1.16 |

Source: (FIORIO; REMES; NOVAES, 2021).

## A.4 FINAL CONSIDERATIONS

The *impulseest* package has been implemented in Python, available at (FIORIO, 2021) and distributed under an open-source MIT license, resulting in a published paper (FIORIO; REMES; NOVAES, 2021). The empirical Bayes method with a QR-matrix computational approach is considered to implement the function. Three regularization kernels are available to the user - DC, TC, and DI - as well as the non-regularized option. The architecture of the software implementation has been kept clean and simple, enabling most Python users to make modifications as desired. Throughout the illustrative examples and experiments, the functionalities of the *impulseest* function are tested. The results show that the use of regularization provided more consistent results, mainly if the TC and DC kernels are considered, leading to lower mean values and standard deviation for the MSE, when compared to other cases.