UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DIRECTORATE OF RESEARCH AND GRADUATE STUDIES

MASTER'S DEGREE IN ELECTRICAL ENGINEERING

MARCO ANTONIO ITABORAHY FILHO

BIO-INSPIRED OPTIMIZATION ALGORITHMS APPLIED TO THE GAPID CONTROL OF A BUCK CONVERTER

DISSERTATION

PONTA GROSSA

2021

MARCO ANTONIO ITABORAHY FILHO

BIO-INSPIRED OPTIMIZATION ALGORITHMS APPLIED TO THE GAPID CONTROL OF A BUCK CONVERTER

ALGORITMOS DE OTIMIZAÇÃO BIO-INSPIRADOS APLICADOS AO CONTROLE DE UM CONVERSOR BUCK

Dissertation presented as a partial requisite to obtain the Master's Degree in Electrical Engineering of the Graduate Program in Electrical Engineering of UTFPR Campus Ponta Grossa.

Advisor: Prof. Dr. Mauricio dos Santos Kaster
Co-Advisor: Prof. Dr. Hugo Valadares Siqueira

PONTA GROSSA

2021

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa

MARCO ANTONIO ITABORAHY FILHO

BIO-INSPIRED OPTIMIZATION ALGORITHMS APPLIED TO THE GAPID CONTROL OF A BUCK CONVERTER

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Controle E Processamento De Energia.

Data de aprovação: 12 de Março de 2021

Prof Mauricio Dos Santos Kaster, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Flavio Trojan, Doutorado - Universidade Tecnológica Federal do Paraná

Prof.a Marcella Scoczynski Ribeiro Martins, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Rafael Stubs Parpinelli, Doutorado - Fundação Universidade do Estado de Santa Catarina (Udesc)

ACKNOWLEDGEMENTS

I would like to thank my family for helping me every step of the way, through thick and thin.

All the Kibes for being the best friends anyone can hope for.

My digital friends from Bingo and Dethroned for the hours of discussion and entertainment during this pandemic.

My professors Hugo and Kaster for their dedication in guiding me.

All my colleagues from LICON for their companionship.

To UTFPR for the support and funding this dissertation.

"There is a theory which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable.

There is another theory which states that this has already happened."

— Douglas Adams

ABSTRACT

ITABORAHY FILHO, Marco Antonio. **Bio-inspired Optimization Algorithms Applied to the GAPID Control of a Buck Converter**. 2021. 59 p. Dissertation (Master's Degree in Electrical Engineering) – Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2021.

The objective of this work is to realize a comparative study of which bio-inspired metaheuristic strategies are the best to be used when finding the optimal gains for a Gaussian Adaptive PID control system applied to a Buck Converter. This comes from two necessities, the first one is that to surpass the performance of linear control systems a more advanced strategy such as adaptive control is required; the second is that since there are no deterministic optimization methods for such strategies, the use of metaheuristics is deemed necessary, and according to the No Free Lunch theorem such comparative studies ought to be made to understand what optimization techniques ares better suitable for such a problem. The chosen techniques were the Genetic Algorithm, the Particle Swarm Optimization and the Differential Evolution. The Gaussian function is employed as the adaptive rule because it is a smooth curve with smooth derivatives that avoids the potential of chattering and abrupt control changes. After evaluating the responses found from simulating all variations of each of the optimization strategies we were able to create an analytical and statistical comparison of each technique and show how an adaptive control system can create a faster and more robust response when compared to the linear PID.

Keywords: Adaptive PID control. Genetic Algorithm. PSO. Differential Evolution.

RESUMO

ITABORAHY FILHO, Marco Antonio. **Bio-inspired Optimization Algorithms Applied to the GAPID Control of a Buck Converter**. 2021. 59 f. Dissertation (Master's Degree em Electrical Engineering) – Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2021.

O objetivo deste trabalho é criar um estudo comparativo de quais estratégias metaheurísticas bioinspiradas podem ser as melhores para otimizar os ganhos ótimos de um sistema de controle PID Adaptativo Gaussiano para um Conversor Buck. Isso vem de duas necessidades, a primeira é que para superar o desempenho dos sistemas de controle linear é necessária uma estratégia mais avançada como o controle adaptativo; a segunda é que como não existem métodos de otimização determinísticos para tais estratégias, o uso de metaheurísticas é considerado necessário, e de acordo com o teorema do *No Free Lunch* tais estudos comparativos devem ser feitos para entender quais técnicas de otimização são mais adequadas para tal problema. As técnicas escolhidas foram o Algoritmo Genético, a Otimização do Enxame de Partículas e a Evolução Diferencial. O Gaussian Adaptive PID (GAPID) foi escolhido para o controle adaptativo por ser ser baseado na função Gaussiana, que é uma curva suave com derivadas suaves que evita vibrações e mudanças bruscas no controle. Depois de avaliar as respostas encontradas na simulação de todas as variações de cada uma das estratégias de otimização, foi criada uma comparação analítica e estatística de cada técnica mostrando como um sistema de controle adaptativo pode gerar uma resposta mais rápida e robusta quando comparado ao PID linear.

Keywords: Adaptive PID control. Genetic Algorithm. PSO. Differential Evolution.

LIST OF FIGURES

LIST OF TABLES

SUMMARY

1 INTRODUCTION

Modern automatic control is essential for any engineering and science fields, from manufacturing systems, robotics, vehicles or any process that involves the control of flow, temperature, humidity, speed. Among the various control techniques available, the PID (proportional, integrative and derivative) stands out, due to its good effectiveness, robustness and well established design methodology; According to Ogata (2001) more than half of the industry relies on PID control systems, indicating the importance of studying such systems.

Despite their popularity in the industry, linear control mechanisms such as PID are constrained by their linearity. Such limitations can be circumvented by employing nonlinear controllers. Optimal controllers include LQR and SDRE, that are based primarily on the Ricatti equation for optimization (TUSSET et al., 2013). Sliding-mode controllers represent a class of discontinuous controllers, with advantages, like fast response and robustness, but also disadvantages, like chattering in the control signal, which limits its application (TAKAHASHI; PERES; BARBOSA, 1999; KASTER; PAGANO, 2006). Adaptive controllers are a special class of such controllers that use adaptive rules to govern the controller behavior (ASTROM; WITTENMARK, 2008). It is very common to employ adaptive rules to the PID controller, to enhance its performance and overcome the limitations of the traditional linear PID. Adaptive control systems such as the GAPID (Gaussian adaptive PID) can generate fast and robust control systems with no loss in quality responses, such as instability against disturbances and plant changes, or high overshoot values.

The use of a GAPID controller was proposed by Kaster et al. (2011), and later on Puchta et al. (2016), introduced the use of bio-inspired metaheuristics for the control optimization, followed by Borges (2019), that included parameters variations in the plant to enhance control robustness.

In this dissertation, the focus is not as much on the plant control such as previous works, but a wider analysis on different metaheuristic optimization algorithms and how they affect the control system's performance.

Such adaptive system are classified as unconstrained continuous optimization problem that do not present deterministic solving methods, thus posing a new problem of finding the best parameters for the controller.

The control parameters that are being optimize are 9 constants for the three Gaussian curves that characterize the adaptive nature of the PID gains of the GAPID (three for each curve), however, with the use of linked parameters (section 2.5) the optimization algorithms have a 6 dimensions problem to solve, rather than a 9 dimensions one.

Some of the most popular mechanisms of non-linear optimization are bio-inspired metaheuristics such as Genetic Algorithms, Differential Evolution and PSO, which will be focused in this work. These strategies were chosen for being some of the most popular metaheuristics in the current research literature and as such, there is a vast bibliography in which to base our studies.

For each of the three optimization strategies used there were 10 different variations applied, resulting in a total of 30 different optimization algorithms, all applied to a GAPID controlling a DC/DC Step Down converter, those optimizations are then simulated and their results are discussed.

1.1 OBJECTIVES

1.1.1 General Objective

The general objective of this dissertation is to utilize different GAPID (Gaussian Adaptive PID) adaptation rules for the control system's parameters for a Buck converter, developing several Bio-inspired Metaheuristic optimization variations of three main strategies that access the step response of the converter.

1.1.2 Specific Objectives

- Design a computational model of a Buck Converter for simulations.
- Investigate the functionality of Adaptive Control strategies.
- Elaborate algorithms for Genetic Algorithms, Particle Swarm optimization, Differential Evolution and their respective variations.
- Run optimization simulations of each metaheuristic variation available and compare the results to the original PID control.
- Study the results obtained and create a comparative analyze their performance utilizing statistical analysis.

- Create documentation and scientific publications based on the research done.

1.2 STRUCTURE OF THE DISSERTATION

This work has the following structure:

Chapter 2 describes the operation of a buck converter, and its usual control strategy, that is the linear PID, the chapter then proceeds to discuss the Gaussian adaptive PID control, how each of its parameters are linked and the performance measurement technique utilized on the optimization process for this paper.

Chapter 3 presents a description of bio-inspired metaheuristics and its advantages over deterministic or heuristic optimization strategies. The chapter then expands on the three metaheuristic strategies that were utilized and their respective variations, those being the Genetic Algorithm, Particle Swarm Optimization and Differential evolution.

Chapter 4 characterizes how each of the simulation experiments were done, what variations and parameters where used and how those results were classified qualitatively utilizing the Fitness function.

Chapter 5 presents the results found after simulating the optimization process for a GAPID controlled buck converter of each of the algorithms presented in the previous chapter, the results for each variation were then analyzed and the best optimization strategies were presented.

Chapter 6 describes the process of researching and developing this work, then presents a few interpretations of the results from Chapter 4 and finishes with a few ideas for the expansion of this paper for the future.

2 GAPID CONTROL

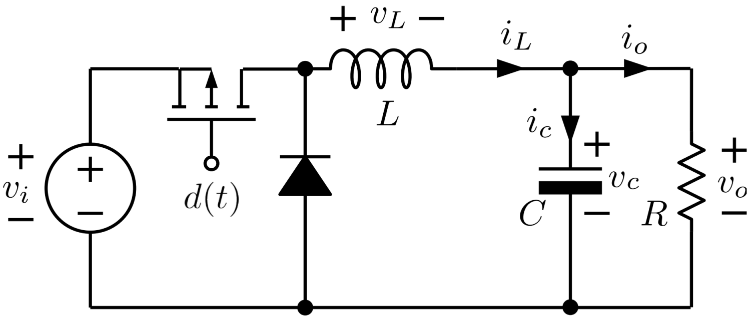
2.1 INTRODUCTION

This chapter will introduce the Application Plant, the Buck Converter, while also introducing the concepts of Linear PID control and its evolution, the Adaptive PID. The concept of linked parameters are also presented and finally the method chosen for measuring the performance of each control strategy.

2.2 BUCK CONVERTER

A Buck converter (Figure 1) is a DC-DC step-down power converter, supplying an output voltage lower than the input voltage and consequently increasing current from input to its load (output) (MANIAS, 2017). This converter topology currently is extremely popular as the main power supply in a large range of electric and electronic equipment, from DC motors to USB chargers.

Figure 1 – Diagram of a Buck Converter



source: Own Authorship, Adapted from Barbi, 2019

where:

$d(t)$ is the state of the switch (1 is on, 0 is off);

v_i is the input DC voltage;

i_L is the current in the inductor;

i_o is the current in the load;

i_C is the current in the capacitor;

L is the inductance value;

C is the capacitance value;

R is the load resistance value;

v_o is the output voltage on the load.

The Buck converter is an electronic system that has the input voltage connected to a MOSFET generating a pulsed voltage that passes by a low-pass filter, represented by the inductor L and the capacitor C, and resulting in a lower output voltage that is used to supply other circuits. The output voltage level can be controlled by varying the duty-cycle $d(t)$ of the PWM signal applied to the MOSFET.

Through nodal analysis it is possible to relate the inductor current as being the sum of the currents going through the capacitor and the load as in equation (1):

$$i_L(t) = i_c(t) + i_o(t) \tag{1}$$

Through mesh analysis, it is possible to determine the input voltage based on equation 2:

$$d(t)V_i = v_L(t) + v_c(t) \tag{2}$$

The discrete function $d(t)$ can be referred as a continuous limited function $u(t)$ that represents the mean of $d(t)$ in one switching period of PWM.

The system can be described by the state-space Equations (3) and (4).

$$L\frac{di_L(t)}{dt} = -v_c(t) + u(t) \cdot v_i \tag{3}$$

$$C\frac{dv_c(t)}{dt} = i_L(t) - \frac{1}{R} \cdot v_c(t) \tag{4}$$

The step-down converter is a linear second-order system. By utilizing the Laplace transform (ROWLEY; BATTEN, 2008) the Transfer Function for the duty-to-output can be obtained, as stated in the equation (5).

$$\frac{V_c(s)}{U(s)} = \frac{V_i}{LCs^2 + \frac{L}{R}s + 1} \tag{5}$$

where $u(t)$ is the PWM function, the control system's input and $U(s)$ its Laplace Transform.

To assure that the converter works as intended, a closed loop control is necessary to stabilize the output voltage at a desired value, which is the main purpose of the converter: to serve as a low-voltage power supply. Actually the most common technique utilized in the industry are linear PID strategies that will be explored in the Section 2.3. In the Section 2.4 the GAPID control is presented as an improvement over the linear PID.

For this project, the Buck converter's desired functionality is to receive a 48 V voltage V_i and to put out the V_o value of 30 V.

2.3 LINEAR PID

The PID is a feedback-based control strategy that uses the error between the plant output and a given set-point to perform the control action, which applies the proportional, integral and derivative rules to this error, resulting the proper control action that cause the plant output to be equal to the desired set-point (HAIDEKKER, 2020).

The control gains are as follow:

g_p, the proportional term involved in the system's gain.

g_i, the integral term that works not only the error, but also the time it has persisted.

g_d, the derivative term doesn't consider the error, but only the change of rate of the error.

For the Buck converter, the output voltage in the the load is controlled by the PWM signal $u(t)$, as stated in the the equation (6)

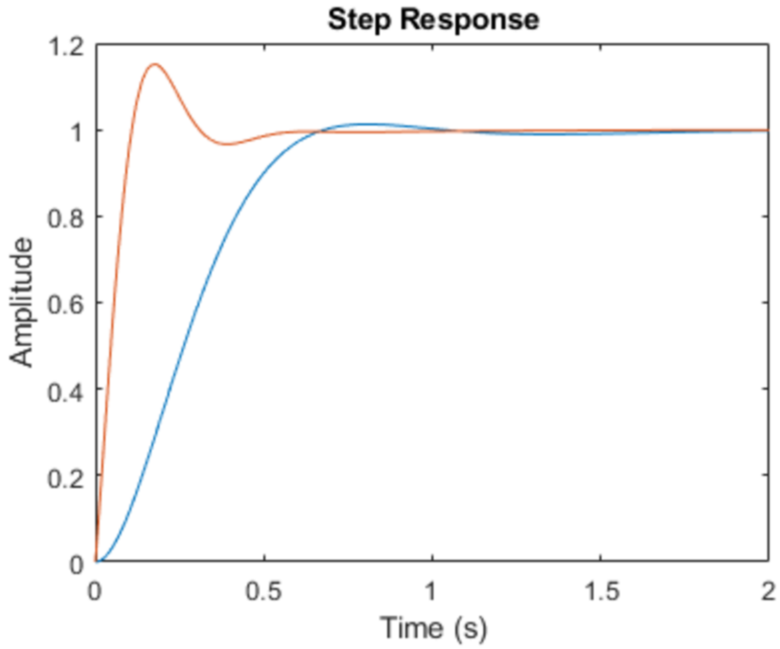
$$u(t) = g_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \tag{6}$$

where $e(t)$is the error signal, T_i is the integral time-constant, T_d is the derivative time-constant, such that $g_i = \frac{g_p}{T_i}$ and $g_d = g_p T_d$.

PID control is widely used in today's industries, its general adaptability and usefulness have inspired the creation of multiple methods of refining and tuning rules such that it's possible to use PID strategies in occasions where the general plant is unknown (OGATA, 2001). However, while linear control methods such as classical PID systems are reliable, there are limitations. The figure 2 presents the comparison of the step responses of two linear PID projects for the same plant and it is an example of such limitations: A PID designed for faster transient responses tends to create overshooting, such as the case of the orange curve. In many cases, the way to reduce that overshoot demands reducing the overall control gain resulting a slower system, as in the blue curve.

The PID limitations are more apparent in complex systems such as those with time-delay, non-linear and time-variant dynamics (HUANG; TAN; LEE, 2001) and researchers have been recurring to adaptive control as better alternatives capable of overcoming such limitations that the simplistic nature of the PID control impose (XU; HOLLERBACH; MA, 1995).

Figure 2 – Buck Converter States

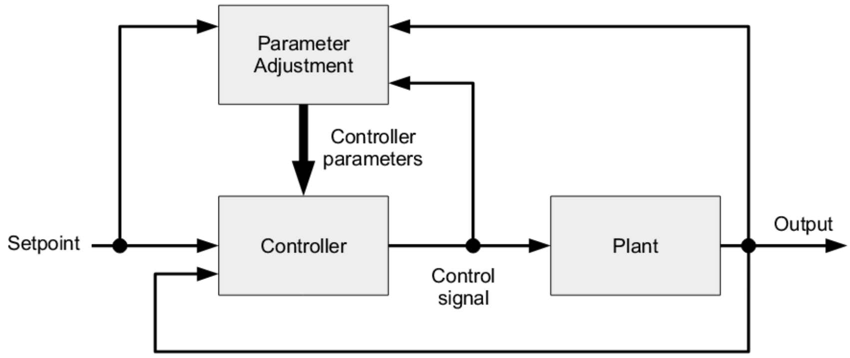


source: Own Authorship

2.4 GAUSSIAN ADAPTIVE PID

An adaptive controller is "a controller with adjustable parameters and a mechanism for adjusting the parameters" (ASTROM; WITTENMARK, 2008) and can be thought of as having two feedback loops, one for the process and one for the parameters, such as presented in the Figure 3. This come with the challenge of finding the adjusting mechanisms or systems that are able to create a better control strategy than the linear one.

Figure 3 – Block Diagram of an Adaptive Control System



source: Adapted from Astrom and Wittenmark, 2008

The GAPID system proposed by Kaster et al. (2011) is an adaptive PID system based on a Gaussian-like function, meaning that each of the PID gains (g_p, g_i and g_d) are obtained through three distinct Gaussian curves based on the current value of set-point/output error. The Gaussian Curves are given by the equations (7), (8) and (9).

$$g_p(\sigma) = k_{p\,1} - (k_{p\,1} - k_{p\,0})e^{-q_p\,\sigma^2} \tag{7}$$

$$g_i(\sigma) = k_{i\,1} - (k_{i\,1} - k_{i\,0})e^{-q_i\,\sigma^2} \tag{8}$$

$$g_d(\sigma) = k_{d\,1}\,(1 - e^{-q_d\,\sigma^2}) \tag{9}$$

where:

σ is the current error

g_p, g_i and g_d are current PID gains based on the error.

$k_{p\,1}$, $k_{i\,1}$,and $k_{d\,1}$, are the gain bound of the Gaussian when error$\rightarrow \infty$ of the Gaussian.

$k_{p\,0}$ and $k_{i\,0}$ are the gain bound of the Gaussian when error$\rightarrow 0$ of the Gaussian.

q_d,q_i and q_d define the degree of concavity of the Gaussian.

As is observable in the equation (9), k_{d0} has been set to zero in order to avoid noise amplification issues in the PID derivative component when the setpoint is reached (Puchta; Siqueira; Kaster, 2020), leaving the GAPID with eight parameters to be designed:

$$[k_{p\,0},k_{p\,1},k_{i\,0},k_{i\,1},k_{d\,1},q_p,q_i,q_d] \in \mathbb{R}^8$$

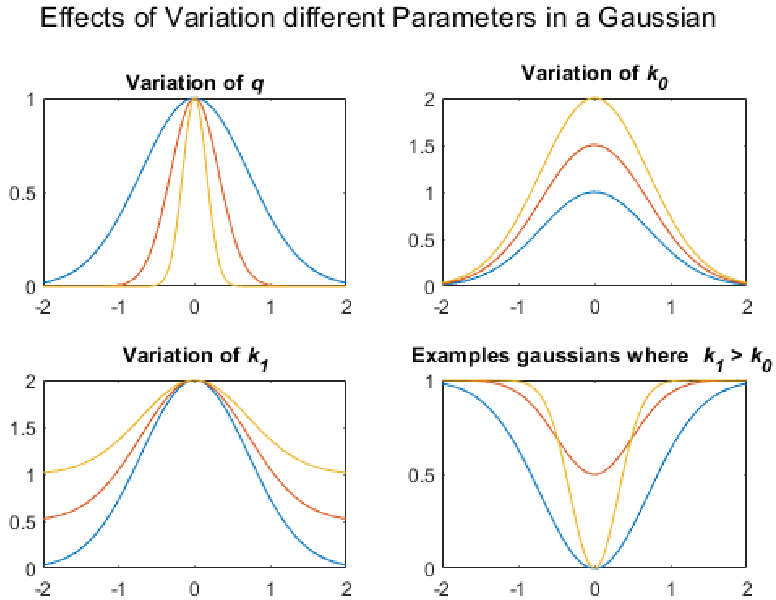
Figure 4 shows examples of how each parameter affects the Gaussian function.

The Gaussian is a smooth function, which allows for a gradual change in the gains, as opposed to discontinuous adaptive functions that has the potential of causing chattering in the points of discontinuity (BORGES, 2019).

The use of the adaptive PID strategy comes with many practical benefits such as the possibility of improving control performance over a linear strategy without losing all the convenience of the well-established support structure behind the linear PID.

Since the GAPID control is based on three Gaussian curves for the values of g_p, g_i and g_d, there are eight independent parameters, as defined in equations (7) to (9), that need to be determined as opposed to the three parameters on the linear PID. Such design represents a hard problem o solve, because there is no mathematical or other deterministic method to determine the parameters. Furthermore it is possible that several distinct solutions can achieve similar performances, wich signals a kind of multimodal problem. In this sense, metaheuristics optimization stands as an important tool to help solving the problem.

Figure 4 – Examples of Gaussian Curves with varying values of k_0, k_1 and q



source: Own Authorship

2.5 LINKED PARAMETERS

As cited previously, on the Section 2.4, the design of GAPID comprises a problem with eight-parameters to be defined. This set of parameters is referred as the Free parameters case, where all GAPID parameters will be considered for optimization.

The problem behind using this approach resides in the possibility of achieving good results, that are optimal in the very specific conditions of the testing computational model, but may perform poorly in real situations. This behavior has been reported by (Puchta; Siqueira; Kaster, 2020) and referred as narrow specialized optimization solution, where the optimized GAPID can perform exceptionally at a specific plant condition but loosing considerable performance upon plant changes and disturbances, meaning that the solution is not robust.

Another issue refers to the appeal to convince industry administrators to adopt a GAPID controller into their processes, some of them with hardly tuned controllers. The fear to subject an expensive industrial process to a new controller, with an unknown tuning process, that can be even harder than the traditional PID, represents an obstacle to the adoption.

Kaster et al. (2011) proposed an alternative approach, by considering the gains of a previously tuned PID controller as a reference to the adaptive gains of the GAPID. In this approach, the upper and lower limits of the Gaussian functions will be derived as direct and inverse relations of each

PID reference gains. So, the relations will be expressed by variables x, y and z for the P-I-D reference gains. It is referred to as the Linked parameters case. This strategy is summarized in equations (10) - (14).

$$k_{p\,1} = x\,k_{p\,PID} \tag{10}$$

$$k_{p\,0} = \frac{1}{x}\,k_{p\,PID} \tag{11}$$

$$k_{i\,1} = y\,k_{i\,PID} \tag{12}$$

$$k_{i\,0} = \frac{1}{y}\,k_{i\,PID} \tag{13}$$

$$k_{d\,1} = z\,k_{d\,PID} \tag{14}$$

Utilizing linked parameters it is possible to also lower the number of variables that need be adjusted to six in the optimization process: [x, y, z, q_p, q_i, q_d]$\in \mathbb{R}^6$ thus reducing the complexity from 8 to 6 dimensions.

Linked parameters was chosen to facilitate the optimization process, by reducing the number of variables, and its increased reliability, by being derived from an already established PID controller.

This chapter presented the application plant, a Buck converter, and discussed the proposed adaptive control strategy, GAPID.

In the next chapter the optimization strategies chosen to use those parameters will be shown in more detail.

3 BIO INSPIRED METAHEURISTICS

3.1 INTRODUCTION

This chapter will introduce the concept of metaheuristics and will develop the ideas of each of the bio-inspired algorithms utilized for unconstrained continuous optimization: Genetic Algorithms, Differential Optimization, and Particle Swarm Optimization.

3.2 BIO-INSPIRED METAHEURISTICS

Most conventional optimization algorithms are deterministic, such as gradient-based methods. However, these methods often fail in cases where there are discontinuities or multiple local optimum (LINDFIELD; PENNY, 2019).

The simplest non-deterministic methods are called Heuristics, meaning “trial and error”. They are not expected to reach the global best result for 100% of the cases. However, it is expected the ability to find fairly good solutions with a limited amount of processing and time.

Further development in the heuristic field were in the metaheuristics methods, with meta meaning “further” or “higher level”. Such methods utilize both randomization and local search techniques, in which the randomization is used as a technique for escaping local optimum points’ attraction basins. Therefore, they also are suitable methods for global optimization (YANG, 2014).

In the current literature there are over 100 metaheuristics strategies and many other variations (YANG, 2014). According to the no Free Lunch (NFL) theorem “any algorithm, any elevated performance over one class of problems is offset by performance over another class” (WOLPERT; MACREADY, 1997), meaning that there is no single algorithm that presents the best performance for every possible optimization problem, one of the reasons for this current study.

For our current problem (the optimization of a GAPID controller of a Buck converter) the chosen metaheuristics methods were three of the most popular strategies found in the literature: Genetic Algorithms (GA), Particle Swarm Optimization (PSO) and Differential Evolution (DE). Their functionalities are presented in this chapter. Also, a candidate solution is generically named as agent.

3.3 LITERATURE REVIEW

The use of metaheuristics for control optimization is of common practice, since most advanced control methods constitute NP-problems that cannot be answered by deterministic approaches, some examples from literature are:

Mendez et al. (2020) utilized the Earthquake Algorithm to optimize the inductance value on a Buck converter to increase the circuit's efficiency to up to 90%.

Jalilvand et al. (2011) was able to tune the PID gains of a DC motor based on PSO optimization.

Ferreira and Coelho (2018) automatized the fine-tuning of a type 2 controller of a Buck converter with Multi-Objective Variable Neighborhood Search.

Alqudah, Malkawi and Alwadie (2014) made use of the Simulated Annealing Optimization method for optimizing the PID gains for control on DC-DC converters.

Puchta et al. (2016) utilized Genetic Algorithms for optimization of the GAPID control of a Buck Converter.

Kaster et al. (2018) created a comparative study of different variations of Genetic Algorithms and their performance when optimizing the control parameters of a Buck Converter through GAPID control.

Borges (2019) expanded the use of not only Genetic Converters, but also PSO optimization for the GAPID control of a Buck converter.

The main contribution of this work is creating a comparative study of a large number of different variations of three main metaheuristic optimization strategies, the Genetic Algorithm, Differential Evolution and Particle Swarm Optimization.

3.4 GENETIC ALGORITHMS

Genetic Algorithms (GA), developed by David E. Goldberg and John H. Holland, are evolutionary search and optimization metaheuristics created with the intent of utilizing the ideas of survival of the fittest elaborated by Charles Darwin in the 19th century (HOLLAND, 1975).

3.4.1 Basic Concepts

The basic concept of a GA is that a group of candidate solutions (agents) are created with a randomly generated set of parameters (genes), creating the population for the first generation. Each agent has its own Fitness value based on its genes. Note that the Fitness is understood as a function that evaluates the solution by giving a value that express how good it is to the current optimization process (MICHALEWICZ, 1996). For our project the genes can be seen as a vector containing the continuous Parameters of the GAPID:

$$[x, y, z, q_p, q_i, q_d] \in \mathbb{R}^6$$

According to Eiben and Smith (2015), each generation is submitted to two main genetic operators: Crossover and mutation, together with a Selection process. The Selection is the procedure that chooses agents for crossover based on their Fitness, creating a higher quality mating pool for

the next generation. In the crossover, the previously selected agents (parents) create an offspring (children) based on their genes. In the mutation, random disturbances from a normal distribution are added to the current gene pool by changing certain genes. The final population is the next generation that will go through that same process until an specific condition is met. A flowchart of the GA process can be observed on the figure 5.

3.4.2 Selection

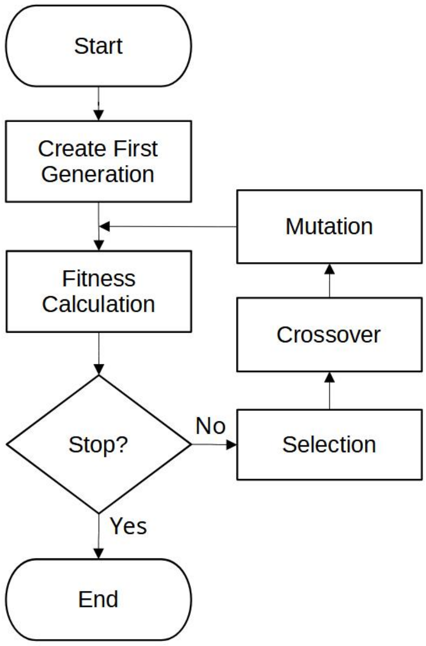
In Genetic Algorithms, the parent selection is the operator responsible to distinguish high fitness agents for mating. The higher the fitness, the higher the probability of an agent be part of the crossover process. The selection is responsible for pushing the quality of the population through probabilistic approaches.

The two most popular categories of selection strategies are roulette wheel and tournament-based (Jinghui Zhong et al., 2005). For roulette-based strategies, the selection is made by picking out agents from a roulette, in which the probability of each agent be picked is directly proportional to its fitness value (figure 6). In tournament-based approaches, the selection is made by comparing two or more random selected agents and choosing the one with highest fitness (figure 7).

The Selection variations chosen in this work are:

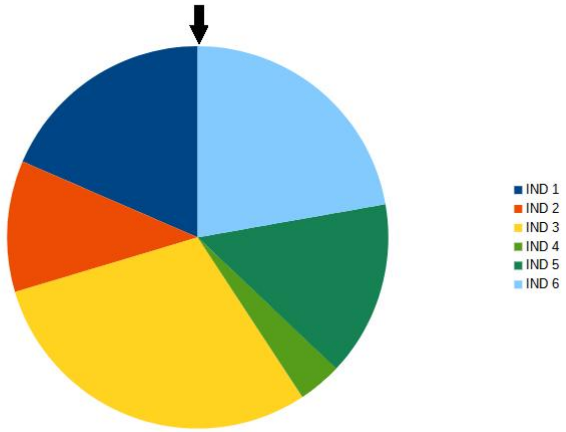
- The classic roulette, in which a number of agents are randomly picked out by using the roulette

Figure 5 – Flowchart of a Genetic Algorithm



source: Own Authorship, Adapted from Holland, 1975

Figure 6 – Example of a roulette-based selection



source: Own Authorship

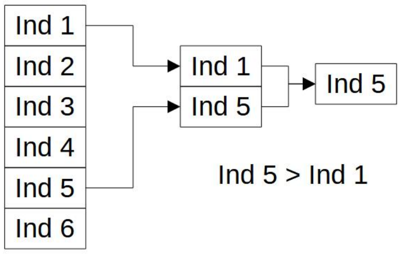
(Jinghui Zhong et al., 2005);

- Stochastic Universal Sampling (SUS), where agents are selected by randomly picking a starting point in the roulette and picking all succeeding agents equally spaced from the previous (PENCHEVA; ATANASSOV; SHANNON, 2010);

- The Classic binary Tournament, with two agents picked randomly and compared. The highest fitness agent wins and gets selected (Jinghui Zhong et al., 2005);

- The elimination tournament, in which after an agent loses at a tournament, it is eliminated from

Figure 7 – Example of a tournament-based selection



source: Own Authorship

the current gene pool (MILLER; GOLDBERG et al., 1995);

- Survival selection that occurs after the crossover. In this case, the selection is made from the population of parents and offspring (Ting; Ko, 2008).

3.4.3 Crossover

The Crossover operator is the equivalent of breeding in nature: two high quality agents capable of surviving the selection process are combined to create a new offspring that takes characteristics from both parents (HOLLAND, 1975). This is the process that creates the new generation.

The Crossover is often completed with a probability usually set to 70% (MARGHANY, 2020) or 100% (MICHALEWICZ, 1996). If it fails, the crossover is skipped, creating agents identical to the parents for the next generation.

The variations of Crossovers chosen for the optimization were:

- Single Point Crossover, where the genetic load of both parents is cut in a random point and then the genes are recombined to create new offspring as shown in figure 8 (HOLLAND, 1975).

- In the arithmetic crossover the offspring is created through the weighted median of both parents, according to the equations (15) and (16) (GOLDBERG, 1994).

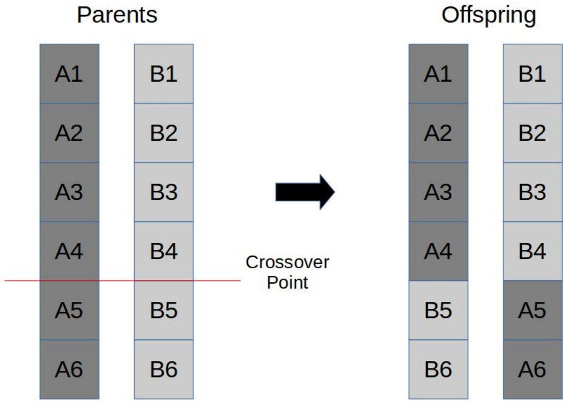
$$\textbf{Offspring1} = \alpha \cdot \textbf{Parent1} + (1-\alpha) \cdot \textbf{Parent2} \tag{15}$$

$$\textbf{Offspring2} = (1-\alpha) \cdot \textbf{Parent1} + \alpha \cdot \textbf{Parent2} \tag{16}$$

where α is a random number between 0 and 1.

- In the SBX Crossover, or Simulated Binary Crossover, each offspring is created simulating the characteristics of a binary crossover. In this case, the average of the parents is equal to the

Figure 8 – Example of a Single Point Crossover



source: Own Authorship

average of the offspring (DEB; KUMAR, 1995). The SBX crossover is given by the expressions (17) and (18).

$$\textbf{Offspring1} = 0.5 \cdot (\textbf{Parent1} + \textbf{Parent2}) - 0.5 \cdot \beta(\textbf{Parent1} - \textbf{Parent2}) \quad (17)$$

$$\textbf{Offspring2} = 0.5 \cdot (\textbf{Parent1} + \textbf{Parent2}) + 0.5 \cdot \beta(\textbf{Parent1} - \textbf{Parent2}) \quad (18)$$

where: $\beta = (2 \cdot u)^{\frac{1}{M+1}}$, if $u < 0.5$ and $\beta = (\frac{0.5}{1-u})^{\frac{1}{M+1}}$, if $u > 0.5$. Note that u is a random number between 0 and 1 and M is a constant usually chosen between 5 and 200.

3.4.4 Mutation

The mutation operator presents the function of adding new information to the current gene pool. Each agent present a chance of having one or more of its genes randomly mutated to a new random value (GOLDBERG, 1989). The probability of occurrence a mutation in an agent is given by the Mutation Rate.

The two mutation variations used were:

- Fixed Mutation rate, in which the mutation rate stays constant through the whole GA process (HOLLAND, 1975).
- Dynamic Mutation, where the mutation rate increases according to the average Fitness value of the population up to a fixed maximum (HONG; WANG, 1996).

3.5 DIFFERENTIAL EVOLUTION

First introduced in 1995 by Reiner Storn and Kenneth Price, the Differential Evolution (DE) was presented as a robust solution for continuous optimization (STORN; PRICE, 1997). DE is also an evolutionary algorithm and its solutions are found using the same concepts of Selection, Crossover and Mutation as in Genetic Algorithms.

3.5.1 Basic Concepts

The Differential Evolution algorithms start, as usual, with a randomly generated population of candidate solutions, called Vectors. The current pool of agents are named Target Vectors. In the mutation step a secondary population is created by adding the position of a Target Vector to the weighted difference of two or more Agents. This mutated agent is called a Donor Vector. For the crossover, a recombination is made utilizing Target and Donor vectors to create a new auxiliary population called Trial Vector. For the Selection, Target and Trial Vectors compete to be the new Target Vector for the next generation (TUSAR; FILIPIC, 2007).

Like the Genetic Algorithms, the agents take the form of the vector:

$$[x,y,z,q_p,q_i,q_d] \in \mathbb{R}^6$$

A simple flowchart of a DE process is shown in figure 9.

3.5.2 Mutation

The mutation can be considered the most important operator in DE. For each Target Vector a new Donor Vector is created by adding one or more differences of random selected Agents. The number of Agents used and which of those Agents will be used determine the type of mutation. All mutation variations can be seen in equations (19) to (23).

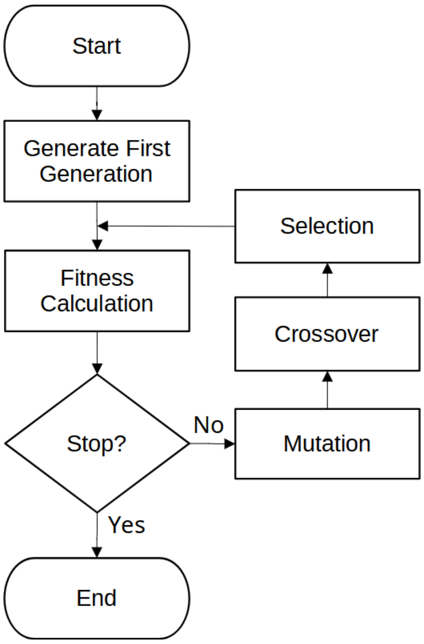
- Rand/1:

$$\boldsymbol{v_i} = \boldsymbol{x_{r1}} + F \times (\boldsymbol{x_{r2}} - \boldsymbol{x_{r3}}) \tag{19}$$

- Best/1:

$$\boldsymbol{v_i} = \boldsymbol{x_{best}} + F \times (\boldsymbol{x_{r1}} - \boldsymbol{x_{r2}}) \tag{20}$$

Figure 9 – Flowchart of a DE



source: Own Authorship

- Rand/2:

$$v_i = x_{r1} + F \times (x_{r2} - x_{r3}) + F \times (x_{r4} - x_{r5}) \tag{21}$$

- Target-to-Best/2:

$$v_i = x_{r1} + F \times (x_{best} - x_{r2}) + F \times (x_{r3} - x_{r4}) \tag{22}$$

- Best/2:

$$v_i = x_{best} + F \times (x_{r1} - x_{r2}) + F \times (x_{r3} - x_{r4}) \tag{23}$$

where:

v_i is the new Donor Vector Agent;

x_{r1}, x_{r2}, x_{r3}, x_{r4} and x_{r5} are Random Target Vector Agents;

x_{best} is the highest fitness Agent on the current Target Vector;

F is a step-size constant, usually set between 0 and 2.

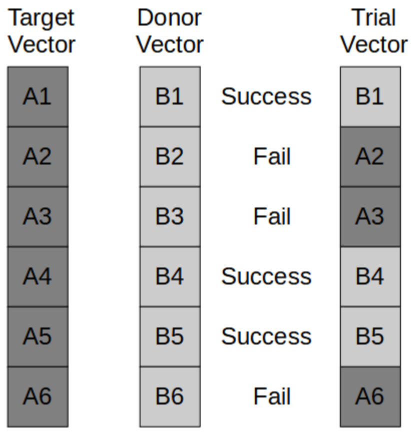
3.5.3 Crossover

The crossover in Differential Evolution creates the Trial Vector by combining both target and donor vectors, mainly with two distinct variations, the binary crossover and exponential crossover (LIN;

QING; FENG, 2011).

On the binary crossover, for each Agent on the Trial Vector, a Bernoulli trial is attempted. The success chance is based on the variable CR: if a trial succeeds, the variable is chosen to fill that position with the Donor Vector. Otherwise the agent position is filled with the Target Vector. An example of this process can be observed on figure 10.

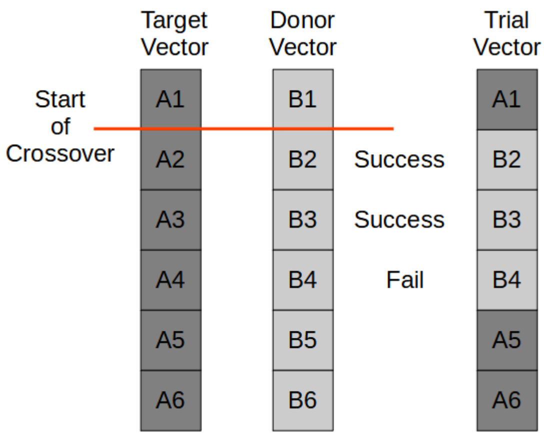
Figure 10 – Example of Binary Crossover



source: Own Authorship

For the exponential crossover, the process starts in a random position on the new trial vector. From that position, the trial vector is filled with the Donor Vector and a Bernoulli trial is attempted (affected by the same factor CR as in the binary crossover). If the trial succeeds, the crossover continues, whoever, the crossover process is ended. An example of an exponential crossover is presented in figure 11.

Figure 11 – Example of Exponential Crossover



source: Own Authorship

3.5.4 Selection

The selection process in differential evolution is relatively simple. It is made by comparing the fitness of agents of both Target Vector and Trial Vector, and selecting the highest fitness Agent, a greedy approach.

3.6 PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO), is a swarm-based optimization technique developed by Kennedy and Eberhart in 1995. The strategy is inspired on the collective behavior of groups of animals, such as school of fish in the open sea or flocks of birds searching for food (KENNEDY; J., 1995).

3.6.1 Basic Concepts

The PSO combines the past social and individual experiences to solve a problem in a collective form. Each agent is named particle, which is able to remember the best solution that it found until the current iteration and to communicate to other particles. The convergence to optimum points happens by means of two forces: the locations of each particle's own best position ($pbest$) and the best solution found by each particle's neighborhood ($gbest$) (KENNEDY; EBERHART; SHI, 2001).

As in the previous optimization strategies, each particle's position is characterized by the vector:

$$[x,y,z,q_p,q_i,q_d] \in \mathbb{R}^6$$

The PSO algorithm starts with a number of candidate solutions being randomly generated. For each succeeding iteration, each particle's fitness value is calculated, and its position for the next iteration is determined by adding its current position to the speed (equations 24 and 25). This is repeated, updating both ($pbest$) and ($gbest$), until a condition is met.

$$\boldsymbol{x}_i(\boldsymbol{t+1}) = \boldsymbol{x}_i(\boldsymbol{t}) + \boldsymbol{v}_i(\boldsymbol{t+1}) \tag{24}$$

$$\boldsymbol{v}_i(\boldsymbol{t+1}) = w \cdot \boldsymbol{v}_i(\boldsymbol{t}) + r_1 \cdot c_1(\boldsymbol{pbest}_i(\boldsymbol{t}) - \boldsymbol{x}_i(\boldsymbol{t})) + r_2 \cdot c_2(\boldsymbol{gbest}(\boldsymbol{t}) - \boldsymbol{x}_i(\boldsymbol{t})) \tag{25}$$

where:

i is the particle.

$\boldsymbol{vi}$ is the velocity of the particle.

w is the inertial factor.

r_1 and r_2 are random numbers between 0 and 1.

c_1 is the personal acceleration factor.

c_2 is the social acceleration factor.

$\boldsymbol{pbest}$ is the best position the particle has been on.

$\boldsymbol{gbest}$ is the best position on a particle's topology.

$\boldsymbol{x_i}$ is the position of the particle at an instant.

The term $c_1(\boldsymbol{pbest_i}(t) - \boldsymbol{x_i(t)})$ is called the cognitive component, affected by the particle's own information; The term $r2 \cdot c2(\boldsymbol{gbest(t)} - \boldsymbol{x_i(t)})$ is the social component, which is affected by the informations from all agents that can communicate with that particle; the variables c_1 and c_2 determine how important is the contribution of each component (BLONDIN, 2009).

The inertial factor w is responsible for increasing or decreasing the particle's speed. A value higher than 1 has the potential to accelerate the particles speeds; a value lower than 1 will always decrease the speed. The value of w can have a lower bound limit because, despite converging faster, the swarm can stagnate the search prematurely and lose the ability of searching for better solutions.

A basic Flowchart of a PSO algorithm can be observed on the figure 12.

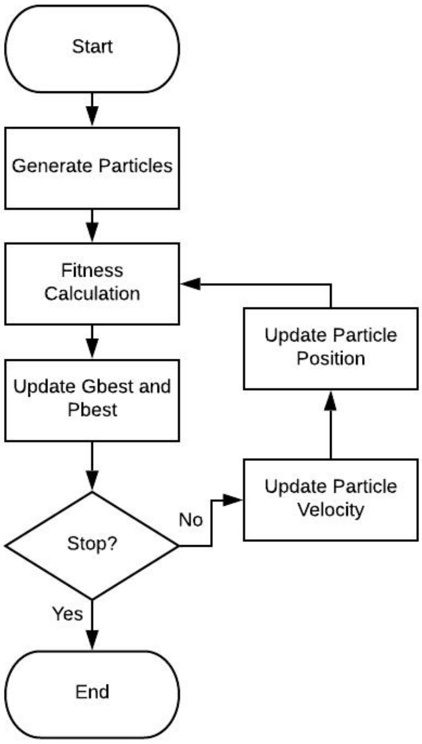
3.6.2 PSO Variants

The first PSO variation explored was based on the topology, mainly the global and ring ones. In the global topology every particle is able to communicate with every other, creating a single shared $\boldsymbol{gbest}$ that all particles tend to follow. In the ring topology, each particle can only communicate with other adjacent particles (PUCHTA et al., 2016). It Is important to note that this adjacency is structural and not based on the particles' positions in the search space. An example of both topologies can be observed on figure 13.

The next approach is the inertial variation, with three options: the *original* method, with a constant value of w and $\boldsymbol{v_i(0)} = 0$; the *initial inertia* method, where the value of w is constant but $\boldsymbol{v_i(0)}$ is randomly generated; in the last case, the *linear decreasing* method, the value of w starts at a higher value. However, over iterations that value decreases while $\boldsymbol{v_i(0)} = 0$ (ALFI, 2011).

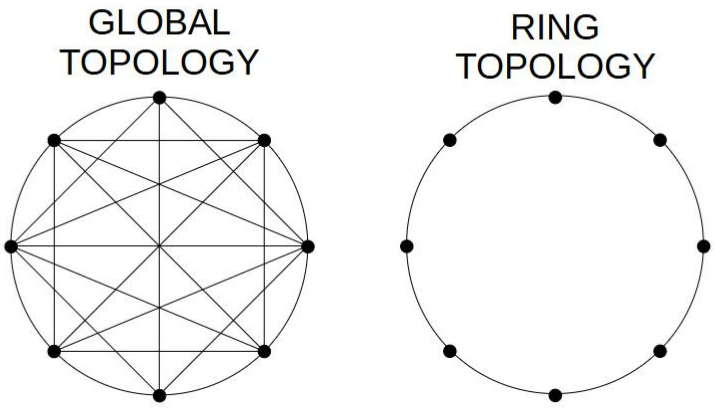
Other variation explores means to escape from local minimum points and spread over the search space. That can be done with two main strategies: the first one is borrowing the mutation

Figure 12 – Flowchart of a PSO



source: Own Authorship

Figure 13 – Example of Global and Ring Topologies for PSO



source: Own Authorship

concept of the GA, where each particle's position can randomly mutate throughout the optimization process (ALFI, 2011); the second strategy is called "bomb", in which in a certain point during the optimization process all particles are submitted to an increase in speed (v_i). This can be seen as a more extreme version of the mutation variation (ALFI, 2011).

This chapter presented the concepts and all the variations of three optimization algorithms employed in this study. In the next chapter, the simulations of all optimization strategies and each variation of the three strategies will be presented and discussed.

4 EXPERIMENTS

4.1 INTRODUCTION

This chapter will stipulate the parameters and variations utilized on the optimization simulations and also the performance measurement function utilized for the qualitative analysis of each of the 30 algorithms.

4.2 PERFORMANCE MEASUREMENT

To evaluate the effectiveness of a set of adjusted gains it is necessary the use of a Performance Measurement method that will be used to compare different results of the optimizations. The standard performance measurement methods are, according to Åström and Hägglund (1995), the IAE or Integral Absolute Error (Equation (26)) and ISE or Integral Square Error (Equation (27)), however ISE applied to PID systems leads to poor gain and phase margins (ZITEK; FIšER; VYHLíDAL, 2016), so the chosen method for our optimizations is the IAE.

$$IAE = \int_0^\infty |\sigma| dt \tag{26}$$

$$ISE = \int_0^\infty \sigma^2 dt \tag{27}$$

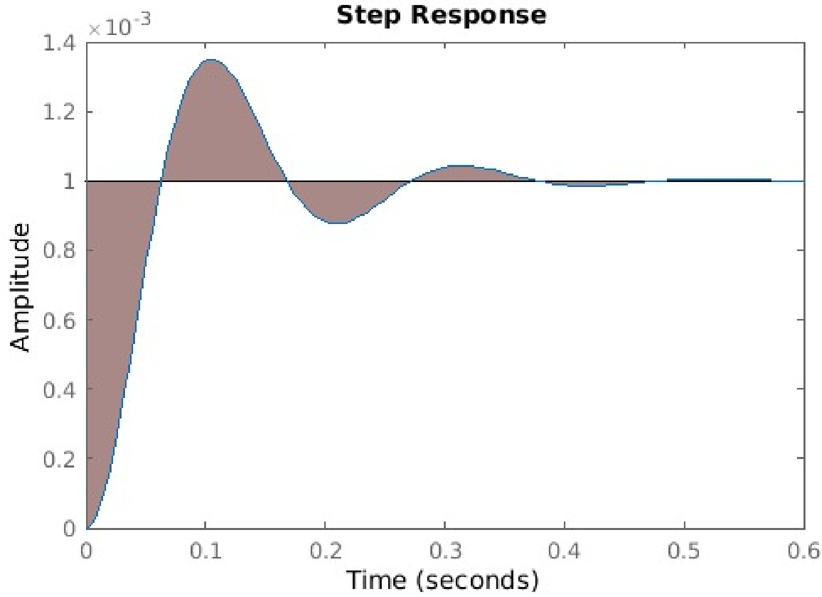
A visualization of the IAE is presented on the figure 14, where the total colored area is the integral of the absolute error.

A fitness function is a performance classification rule that is used in optimization that allows algorithms and humans to better understand the data from a process. In this sense, the highest fitness value given by a determined candidate, the better suited that candidate is to solve the problem.

The fitness function utilized for the optimization of the buck converter was created for the minimization of the error as a result of a step input on the plant and can be observed on the equation (28), facilitating the observation of the Fitness value, since it is now a normalized value between 0 and 1 where the closer to the unit it is, the better.

$$Fitness = \frac{1}{1 + IAE} \tag{28}$$

Figure 14 – Example of a IAE



source: Own Authorship

4.3 TRANSIENT ANALYSIS

To verify a system's performance, it is very usual to employ a reference change, which means an input step. The resulting transient presents several important characteristics, where two are of special interest in our case: the settling-time and the overshoot. By tanking into account only the output signal, the best case is represented by a transient with the lowest overshoot and settling-time.

As stated previously in section 4.2, the IAE index represents a good performance indicator, which balances well the overshoot and settling-time.

4.4 USED PARAMETERS

The following results were obtained considering the buck converter and the GAPID control, which were simulated in the platform Matlab/Simulink© utilizing a personal computer with an i7-6700HQ, 2.6 GHz CPU with 8 cores, an NVIDIA GeForce GTX 960M GPU, and 16 GB RAM. Each optimization variation was simulated 30 times, since, according to Hogg and Tanis (2010), this is a sample size that creates statistical relevance to the performance analysis that was done.

The parameters used for each optimization strategy were:

- For Genetic Algorithms: a population of 50 individuals and 30 generations;

- For Differential Evolution: a population of 50 agents and 100 iterations;
- For PSO: a population of 50 particles and 30 iterations.

The difference between the number of iterations from the DE and the other strategies comes from the crossover rate CR of 0.3, since it is possible to generate a higher number of iterations with the same computing power than other strategies.

Other parameters, defined empirically, were: for genetic algorithms, the mutation rate was set to 5% in all variations but the dynamic mutation to one, in which the mutation rate varied from 1% to 30% depending on the mean fitness value of the population. A value of $M = 50$ for the SBX crossover on Genetic Algorithms, for DE, the value of F was set as 0.8; for the PSO, in strategies with constant inertia values, $w = 0.5$, while in the the decreasing inertia the value of w varied between 0.1 and 0.9.

4.5 GENETIC ALGORITHMS VARIATIONS

The ten variations of genetic algorithms chosen for optimization were:

1. Roulette, Single-point crossover, 70% crossover rate, fixed mutation rate;
2. Roulette , Single-point crossover, 100% crossover rate, fixed mutation rate;
3. Roulette with Survival Selection, Single-point crossover, 100% crossover rate, fixed mutation rate;
4. Tournament, Single-point crossover, 100% crossover rate, fixed mutation rate;
5. Tournament with Survival Selection, Single-point crossover, 100% crossover rate, fixed mutation rate;
6. Death Tournament, Single-point crossover, 100% crossover rate, fixed mutation rate;
7. Stochastic sampling, Single-point crossover, 100% crossover rate, fixed mutation rate;
8. Tournament, arithmetic crossover, 100% crossover rate, fixed mutation rate;
9. Tournament, Single-point crossover, 100% crossover rate, dynamic mutation rate;
10. Tournament, SBX crossover, 100% crossover rate, fixed mutation rate.

4.6 DIFFERENTIAL EVOLUTION VARIATIONS

Differential Equations have a specific nomenclature pattern that facilitates the identification of which strategy is being utilized. That pattern is “Type of mutation / Number of agent pairs used / Type of crossover”. For example, a variation that utilizes a mutation with random Agents, two weighted agent differences and an exponential crossover would be named “Rand/2/Exp” (TUSAR; FILIPIC, 2007).

All the DE variations utilized for the optimization were:

1. Rand/1/Bin
2. Best/1/Bin
3. Rand/2/Bin
4. Target-to-Best/2/Bin
5. Best/2/Bin
6. Rand/1/Exp
7. Best/1/Exp
8. Rand/2/Exp
9. Target-to-Best/2/Exp
10. Best/2/Exp

4.7 PSO

The PSO variations chosen for the GAPID simulations were:

1. No initial inertia, global topology
2. Initial Inertia, global topology
3. Decreasing Inertia, global topology
4. No initial inertia with mutation, global topology

5. No initial inertia with a bomb, global topology
6. No initial inertia, ring topology
7. Initial Inertia, ring topology
8. Decreasing Inertia, ring topology
9. No initial inertia with with mutation, ring topology
10. No initial inertia with a bomb, ring topology

The next chapter discusses the results found on the experiments presented on this chapter.

5 RESULTS

This chapter shows the preliminary results achieved by the optimization algorithms for each variation previously presented. First it was calculated the median fitness of each different optimization variations and the standard deviation for each one, and also created a boxplot of those values.

Then those results were analyzed utilizing the Shapiro Wilks test for parametrization, the Kruskal-Wallis Test for comparing which samples are independent, the Dunn-Sidak correction for a familywise test of the variations.

At the end, the graphics of voltage, error, and gain for the best set of parameters found for each strategy and a brief discussion of the results found are made.

5.1 DESCRIPTIVE ANALYSIS

Utilizing the data that was acquired following the procedures cited in chapter 4, it was possible to create the tables 1, 2, and 3 that show the best Fitness values found for each strategy and the median fitness together with the Standard Deviation.

Table 1 – Fitness values for different Variations of GA

GA Variation	Highest Fit.	Median Fit.	Standard Deviation
GA 1	0.9934	0.9904	0.0021
GA 2	0.9927	0.9909	0.0017
GA 3	0.9927	0.9891	0.0029
GA 4	0.9935	0.9917	0.0024
GA 5	0.9930	0.9880	0.0031
GA 6	0.9933	**0.9917**	0.0015
GA 7	0.9933	0.9904	0.0020
GA 8	0.9915	0.9896	0.0015
GA 9	0.9929	0.9906	0.0022
GA 10	**0.9940**	0.9915	**0.0012**

Starting by GA, as It's possible to see, the best solution found by the Genetic Algorithms was GA 10 (Tournament, SBX crossover, fixed mutation rate) while the highest mean variation was GA 6 (Death Tournament, Single-point crossover, fixed mutation rate)

For PSO, the best solution was from PSO 2 (initial Inertia, global topology), both in terms of the highest fitness value and the highest mean found.

For Differencial Evolution, the highest fitness was reached by the variation DE 2 (Best/1/Bin) while the best mean came from DE 4 (Target-to-Best/2/Bin).

Table 2 – Fitness values for different Variations of PSO

PSO Variation	Highest Fit.	Median Fit.	Standard Deviation
PSO 1	0.9946	0.9920	0.0021
PSO 2	**0.9951**	**0.9947**	0.0012
PSO 3	0.9951	0.9935	0.0012
PSO 4	0.9946	0.9935	**0.0006**
PSO 5	0.9947	0.9934	0.0013
PSO 6	0.9937	0.9915	0.0012
PSO 7	0.9943	0.9939	0.0013
PSO 8	0.9947	0.9921	0.0013
PSO 9	0.9941	0.9919	0.0007
PSO 10	0.9940	0.9926	0.0011

Table 3 – Fitness values for different Variations of DE

DE Variation	Highest Fit.	Median Fit.	Standard Deviation
DE 1	0.9943	0.9935	0.0009
DE 2	**0.9965**	0.9934	0.0009
DE 3	0.9947	0.9932	0.0007
DE 4	0.9962	**0.9936**	0.0016
DE 5	0.9962	0.9930	0.0012
DE 6	0.9933	0.9925	0.0010
DE 7	0.9956	0.9921	**0.0007**
DE 8	0.9934	0.9922	0.0010
DE 9	0.9947	0.9926	0.0012
DE 10	0.9954	0.9918	0.0013

Based of those same results, the Boxplot graph regarding the fitness from each algorithm that resulted from the simulations can be observed in Figure 15 where it is possible to have a broad view of how each one of the variations behaved during optimization.

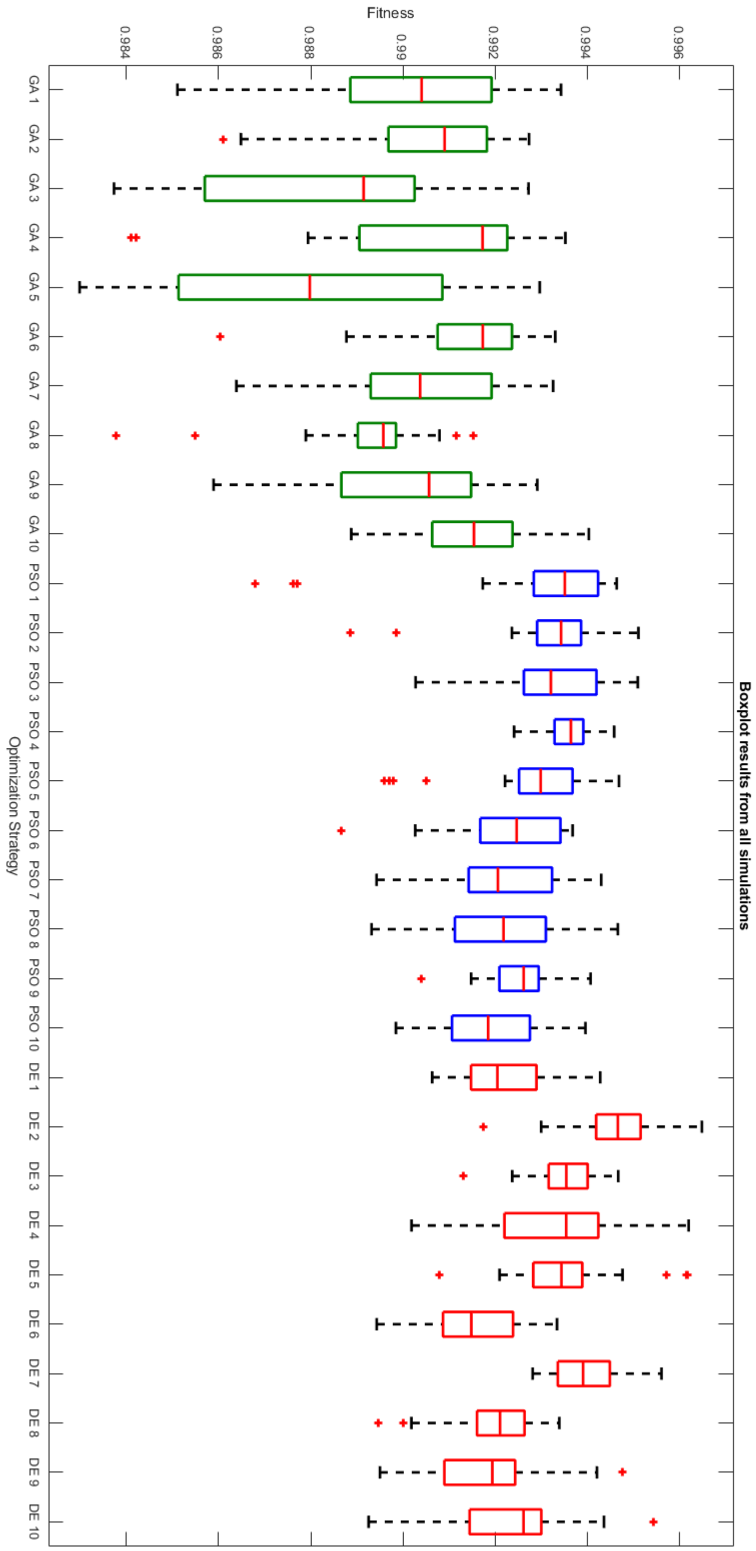
5.2 STATISTICAL TESTS

The results of the simulations were submitted through a series of statistical tests with the intent of analyzing the differences between each optimization method. Each test has a Null Hypothesis, also known as the conjecture, that tries to be disproven (or rejected) by mathematical and statistical analysis.

5.2.1 Shapiro Wilk Test

Before further analysis, the results must be categorized in regards of parametrization. Each collected data was tested using the Shapiro-Wilk method, a test of normality used in statistics that can determine if a sample of numbers came from a normal distribution (SHAPIRO; WILK, 1965).

Figure 15 – Boxplot of Fitness Values found for each optimization strategy



source: Own Authorship

For this test, the null hypothesis is: The sample comes from a normal distribution with unspecified mean and variance. For this test the standard significance level of 5% was used, meaning that every p-value found lower than 0.05 does reject the null hypothesis and does not come from a normal distribution.

The formula for the Shapiro Test can be observed in the equation (29).

$$W = \frac{(\sum_{i=1}^{n} a_i \cdot x_{(i)})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{29}$$

where:

$$(a_1, \cdots, a_n) = \frac{m^T \cdot V^{-1}}{C} \qquad C = \left\|V^{-1} \cdot m\right\| = (m^T \cdot V^{-1} \cdot V^{-1} \cdot m)^{1/2}$$

The vector $m = (m_1, \cdots, m_n)^t$ is created from the expected values of the order statistics of independent and identical distributed random variables sampled from a standard normal distribution.

V is the co-variance matrix of those normal order statistics.

And $\bar{x}$ is the sample mean.

The test results for all samples from the simulations of each strategies' variations are shown on tables 8, 9 and 10 on Appendix A.

Utilizing the fact that in the Shapiro Wilk tests a large part of the results were deemed non-parametric (GA 2, GA 3, GA 4, GA 6, GA 8, PSO 2, DE 1, DE 2, DE 4 AND DE 6) it is possible to treat all samples as non-parametric for the future tests.

5.2.2 Kruskal–Wallis test

The second analysis was done through the Kruskal–Wallis test, a "non-parametric method for testing whether samples originate from the same distribution" (KRUSKAL; WALLIS, 1952). The null hypothesis of the test is: The mean ranks of each groups are the same, or come from populations of the same distribution. A rejection of the null-hypothesis would signify that the different variations tested do indeed create independent results, and therefore are meaningful for this study.

The formula for the Kruskal-Wallis test can be observed on the equation (30).

$$H = (N-1) \cdot \frac{\sum_{i=1}^{g} n_i (\bar{r}_i - \bar{r})^2}{\sum_{i=1}^{g} \sum_{j=1}^{n_i} (r_{ij} - \bar{r})^2} \tag{30}$$

where:

n_i is the number of observations in the ith group;

r_{ij} is the rank (among all observations) of observation j from group i;

N is the total number of observations across all groups;

$\bar{r}_i = \frac{\sum_{j=1}^{n_i} r_{ij}}{n_i}$;

and $\bar{r} = \frac{1}{2} \cdot (N+1)$ is the average rank of all observations on the ith group.

After applying the Kruskal-Wallis test to the data found from each variation of all three optimization strategies the following p-values were found:

- For Genetic Algorithms: $9.566 \cdot 10^{-11}$
- For Differential Evolution: $3.0844 \cdot 10^{-28}$
- For PSO: $1.379 \cdot 10^{-11}$

Considering the standard rejection p-value of 0.05, it can be said that the variations applied to each strategy were, indeed, a differential factor for the GAPID control performance and the results independent from each other.

5.2.3 Dunn-Sidak correction

The third statistical test applied was the Dunn-Sidak correction Post-Hoc analysis, a family-wise error test that corrects for multiple comparisons. The equation (31) presents the test where, given k different null hypothesis of α significance level, each null hypothesis that has a p-value under α^* will be rejected.

$$\alpha^* = 1 - (1-\alpha)^{\frac{1}{k}} \tag{31}$$

The Null hypothesis of this test is: For each pair comparison, the probability of a randomly selected value from one group be larger than a randomly selected value from the second group equals one half, and this can be understood as a median comparison test. When rejecting the null hypothesis, the test stipulates a significant difference between those two results.

To determine statistically the best variations for each strategy, the variations that presented highest Fitness values (GA 10, ED 2 and PSO 2) were compared against all other variations from the same strategy.

The results for those tests can be observed on the tables 4, 5 and 6.

Table 4 – Dunn-Sidak test results for GA 10

GA Variation	Q value	Crit. Q	Null-hypothesis
GA 1	No comparison made	3.2537	Accept Ho
GA 2	No comparison made	3.2537	Accept Ho
GA 3	5.2505	3.2537	Reject Ho
GA 4	No comparison made	3.2537	Accept Ho
GA 5	4.9960	3.2537	Reject Ho
GA 6	No comparison made	3.2537	Accept Ho
GA 7	No comparison made	3.2537	Accept Ho
GA 8	4.7385	3.2537	Reject Ho
GA 9	2.7636	3.2537	Fail to reject Ho
GA 10	—	3.2537	—

Table 5 – Dunn-Sidak test results for ED 2

ED Variation	Q value	Crit. Q	Null-hypothesis
ED 1	6.6256	3.2537	Reject Ho
ED 2	—	3.2537	—
ED 3	2.7324	3.2537	Fail to reject Ho
ED 4	3.3917	3.2537	Reject Ho
ED 5	No comparison made	3.2537	Accept Ho
ED 6	8.3103	3.2537	Reject Ho
ED 7	No comparison made	3.2537	Accept Ho
ED 8	7.1971	3.2537	Reject Ho
ED 9	7.5528	3.2537	Reject Ho
ED 10	6.1434	3.2537	Reject Ho

Table 6 – Dunn-Sidak test results for PSO 2

PSO Variation	Q value	Crit. Q	Null-hypothesis
PSO 1	No comparison made	3.2537	Accept Ho
PSO 2	—	3.2537	—
PSO 3	No comparison made	3.2537	Accept Ho
PSO 4	No comparison made	3.2537	Accept Ho
PSO 5	No comparison made	3.2537	Accept Ho
PSO 6	3.1610	3.2537	Fail to reject Ho
PSO 7	3.5316	3.2537	Reject Ho
PSO 8	3.7161	3.2537	Reject Ho
PSO 9	3.2949	3.2537	Reject Ho
PSO 10	4.4468	3.2537	Reject Ho

As observed, the best Genetic Algorithm variation GA 10 (Tournament, SBX crossover, 100% crossover rate, fixed mutation rate) did not reject the null hypothesis on the cases for GA 1 (Roulette with a 70% crossover rate, Single-point crossover, fixed mutation rate), GA 2 (Roulette, Single-point crossover, fixed mutation rate), GA 4 (Tournament, Single-point crossover, fixed mutation rate), GA 6 (Death Tournament, Single-point crossover, fixed mutation rate) and GA 7 (Stochastic sampling, Single-point crossover, fixed mutation rate) and GA 9 (Tournament, Single-point crossover, dynamic mutation rate), meaning that they are statistically equivalent.

For the Particle Swarm Optimization, the variations the PSO1 (No initial inertia, global topol-

Table 7 – Dunn-Sidak test results for DE 2 compared to other strategies

Variation	Q value	Crit. Q	null hypothesis
PSO 1	No comparison made	3.8504	Accept Ho
PSO 2	No comparison made	3.8504	Accept Ho
PSO 3	No comparison made	3.8504	Accept Ho
PSO 4	No comparison made	3.8504	Accept Ho

ogy), PSO3 (Decreasing Inertia, global topology), PSO 4 (No initial inertia with with mutation, global topology), PSO 5 (No initial inertia with a bomb, global topology) and PSO 6 (No initial inertia, ring topology) variations were considered to be statistically similar to the best candidate PSO 2 (Initial Inertia, global topology).

And for the DE algorithm with the best response overall, DE 2 (Best/1/Bin), the Null Hypothesis was accepted for: DE3 (Rand/2/Bin), DE 4 (Target-to-Best/2/Bin), DE 5 (Best/2/Bin) and DE 7 (Best/1/Exp), and, when comparing it to variations of other strategies, PSO 1 (No initial inertia, global topology), PSO2 (Initial Inertia, global topology) PSO 3 (Decreasing Inertia, global topology) and PSO 4 (No initial inertia with with mutation, global topology) were also considered to be sufficiently comparable, while no GA variation was able to match that fitness value as observable on the table 7.

5.3 RESPONSE TO THE STEP

The response to the step input of 30V of the best set of parameters found by the Genetic Algorithm, Differential and PSO and also the original PID control in which these parameters where derived from (see Section 2.5) can be observed in figure 16.

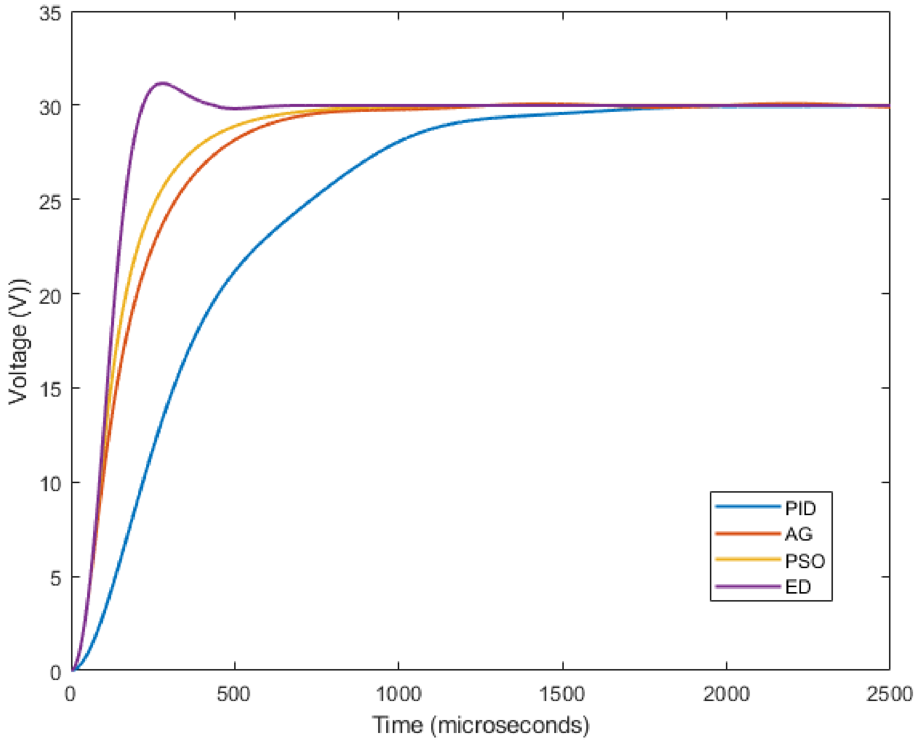
As it's possible to observe, there is a clear improvement on the performance of the system without overshoot for both Genetic Algorithm and PSO strategies, while the Differential Evolution's response generated a response that was significantly faster than those strategies with a small overshoot value.

It is also possible to observe the error between the Buck Converter's voltage and the set-point of 30 V on figure 17 where one can see how the lower error values would create lower IAE values, and consequently, higher Fitness.

The value of the PID gains (g_p, g_d and g_i) of those parameters found during the transient period of the GAPID are displayed on figures 18, 19 and 20.

It is possible to observe how, during the optimization process, the variance caused by the adaptive nature of the GAPID causes the proportional and derivative gains decrease as the error

Figure 16 – Result of the best fitness values found in each optimization strategy applied to the GAPID controller compared to the original PID control



source: Own Authorship

becomes zero, leaving the system to be controlled solely by the integral component when the Buck converter's output reaches the steady-state.

Another signal that can be analyzed is the effort spent by the control action, that can be observed on figure 21, it is possible to notice that the adaptive control systems are faster due to the higher control action right at the beginning of the step, while the pure PID system is slower right at the start.

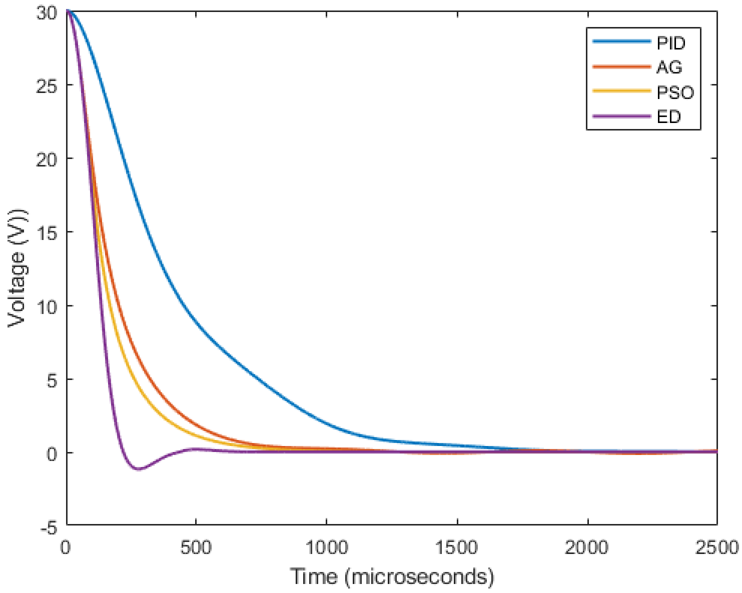
5.4 RESULTS ANALYSIS

Analyzing the Boxplots, the Shapiro-Wilks, Kruskal-Wallis and Dunn-Sidak test results and the response to a step signal, it is possible to come up with some general observations:

For GAs:

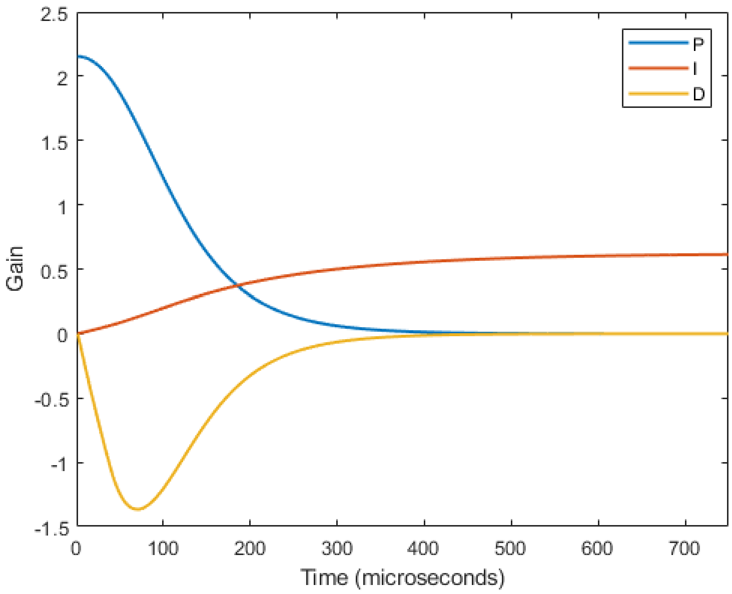
- Roulette-based strategies (GA 1, and 2) have an overall worse performance when compared to Tournament strategy (GA 4);
- The Stochastic Sampling selection (GA 7) had lower Fitness values when compared to the

Figure 17 – Error values found in each optimization strategy applied to the GAPID controller compared to the original PID control



source: Own Authorship

Figure 18 – Values of the gains g_p, g_d and g_i for the best GA 10

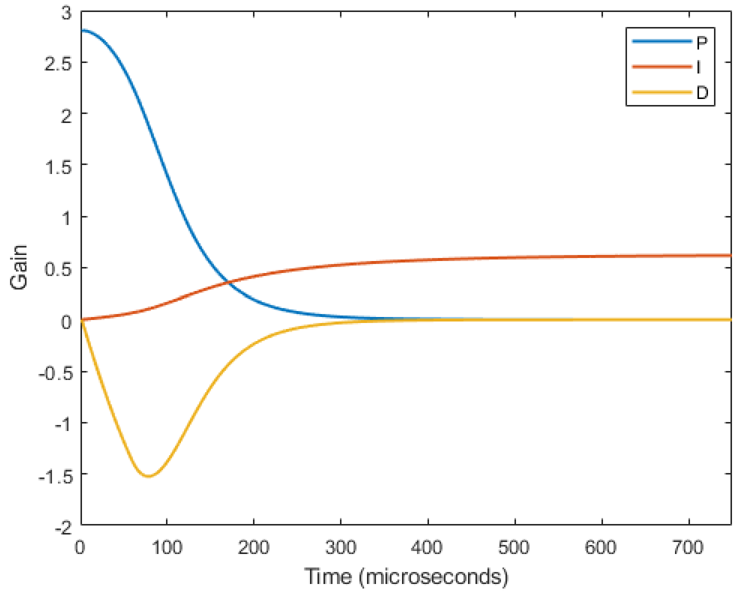


source: Own Authorship

Roulette strategy (GA 2) and Tournament (GA 4);

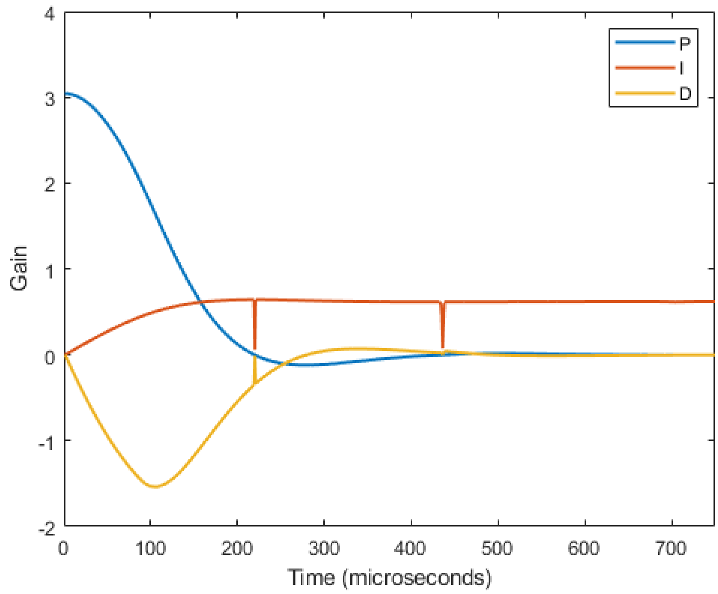
- The Survival Selection (GA 3 and 5) was not beneficial for this optimization problem;
- The Death Tournament (GA 6) resulted in a lower standard variation when compared to the regular Tournament (GA 4);

Figure 19 – Values of the gains g_p, g_d and g_i for the best PSO 2



source: Own Authorship

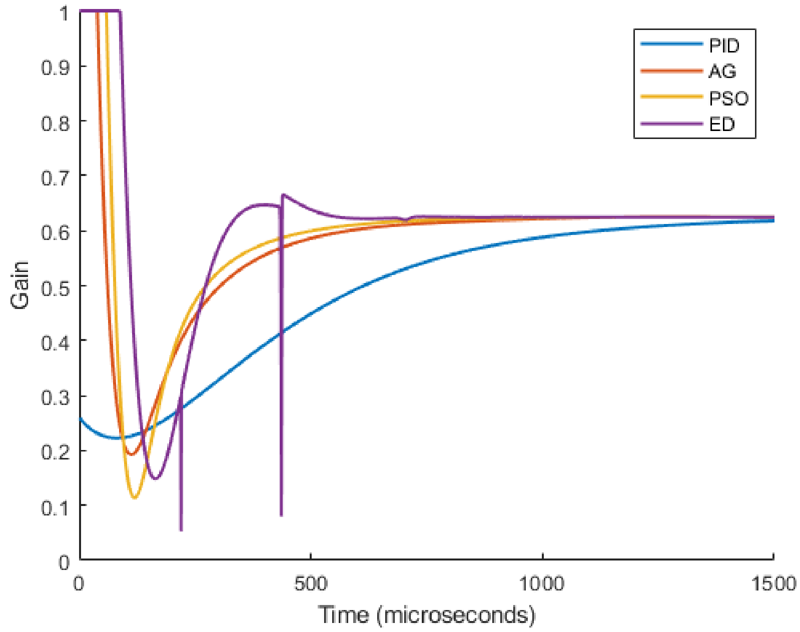
Figure 20 – Values of the gains g_p, g_d and g_i for the best DE 2



source: Own Authorship

- The Arithmetic Crossover (GA 8) achieved was not able to find high quality parameters;
- The Dynamic mutation (GA 9) had an increased the spread but with no real performance gain;
- The SBX crossover (GA 10) obtained the best results for the tested GA variations. However, analyzing the Dunn-Sidak results this strategy is statistically similar to the strategies GA 1, GA 4, GA 6 and GA 7.

Figure 21 – Comparison of control effort from the PID and each of the best GAPID strategies



source: Own Authorship

For PSO:

- Global Topology (PSO 1 through 5) performed better than their Ring Topology counterparts (PSO 6 through 10);
- Both Decreasing Inertia strategies (PSO 3 and PSO 7) offered the second and third lowest PSO errors, respectively;
- The Mutation variation (PSO 4 and 9) lowered the results spread and it's optimizations were, on average, better than the other variations;
- The bomb variation (PSO 5 and 10) did not appear to have improved the performance of optimization;
- The Initial Inertia variation with Global Topology (PSO 2) offered the highest fitness values for PSO strategies, but analyzing the Dunn-Sidak correction, this variation can be considered tied in efficiency to PSO 5 and PSO 7.

For DEs:

- Overall, Differential Evolution strategies achieved lower error values than both GAs and PSO strategies;

- The Binary Crossover (DE 1 to 5) generally achieved better agents when compared to the exponential crossover (DE 6 to 10);
- Rand/1 type strategies (DE 1 and 6) showed lower performance compared to other DEs;
- Best/1 variations (DE 2 and 7) was the best Mutation variation found on DE high quality results;
- Rand/2, Target-to-best, and Best/2 variations produced, overall, low Fitness individuals compared to the best strategies;

The best metaheuristic variation found through the simulations for all three strategies was the DE 2. However, the Dunn-Sidak results indicate that that variation is statistically tied in performance to the variations DE 3, DE 4, DE 5, DE 7 as well as PSO 1, PSO 2, PSO 3 and PSO 4.

6 CONCLUSIONS

This work started with the realization that there was no broad comparative study about different metaheuristic techniques applied to plants such as the Buck converter, and that it would be the subject of research in this work.

The next step was the study of the most relevant bibliography about Buck converters, linear and adaptive control and metaheuristic optimization, more importantly GA, DE and PSO.

Then, working models of a system with the GAPID controlling the Buck converter was created on the platform Matlab/Simulink©. The algorithms for each of different variations of GAs, PSOs and DEs were developed and applied to the optimization of the GAPID controller parameters, following the rules described in the chapter 5.

After the simulations were concluded, the resulting responses and fitness values of each simulation were recorded and analyzed and presented in the chapter 5. A few conclusions were possible to be done.

The fitness function in equation (28) proved to be a suitable choice for the optimization considering for the step response, converging on results of low settling-time, as presented on figure 16, and the highest Fitness Value strategies were responsible for decrease the settling-times of the GAPID.

The GAPID control with Linked Parameters showed a significant improvement over the linear PID control (figure 16), and the Friedman Tests demonstrated that a different variation of the three main strategies were responsible for creating independent responses.

The Differential Evolution strategies were overall better suited for optimization, followed by the PSO and the Genetic Algorithms. Although, all strategies and variations were able to improve the step response over the original linear PID.

After the statistical analysis considering the Dunn-Sidak correction, the best Genetic Algorithm variations found were: GA 1, GA 4, GA 6, GA 7 and GA 10. For Particle Swarm Optimization, the best variation were the PSO 2, PSO 5 and PSO 7 and for the Differential Evolution the best variations were the DE 2, DE 3, DE 4, DE 5 and DE 7.

The overall best strategies variations were PSO 1, PSO 2, PSO 3, PSO 4, DE 3, DE 4, DE 5 and DE 7, while DE 2 presented the highest Fitness. According to the Dunn-Sidak correction there's no statistical difference among them and the other strategies cited.

As a result of this study, the following paper has been submitted to an open-access journal and is under analysis as of the date of the final realease of this dissertation:

- PUCHTA, Erickson D. P.; BASSETO, Priscilla; BIUK, Lucas H.; ITABORAHY FILHO, Marco A.; KASTER, Mauricio S.; SIQUEIRA, Hugo V. **Swarm-Inpired Algoritms Applied to Optimize Gaussian Adaptive PID Controller**. MDPI Energies Open-access Journal. ISSN: 1996-1073. Submitted on april/2021.

As a continuation of this investigation, further works can be proposed:

- Explore other metaheuristic algorithms in this application and make a wider analysis including a broader range of algorithms.
- Make comparisons of metaheuristic optimization processes against deterministic ones.
- Perform robust optimization by taking into consideration variable parameters, like the output load, and derive solutions that perform best under a specified range of variable parameter conditions.
- Investigate strategies to perform optimization on-line, where in-place controllers should be able to tune themselves while the process is running.
- Apply this control technique and optimization to other different plants, including nonlinear ones.
- Apply multi-objective optimization techniques.

REFERENCES

ALFI, Alireza. PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems. **Acta Automatica Sinica**, Elsevier BV, v. 37, n. 5, p. 541–549, maio 2011. quoted 2 time on pages 31 and 32.

ALQUDAH, Amin; MALKAWI, Ahmad; ALWADIE, Abdullah. Adaptive control of DC-DC converter using simulated annealing optimization method. **Journal of Signal and Information Processing**, Scientific Research Publishing, Inc., v. 05, n. 04, p. 198–207, 2014. Disponível em: <https://doi.org/10.4236/jsip.2014.54021>. Quoted on page 22.

ÅSTRÖM, Karl Johan; HÄGGLUND, Tore. **PID Controllers: Theory, Design, and Tuning**. [S.l.]: ISA - The Instrumentation, Systems and Automation Society, 1995. ISBN 1-55617-516-7. Quoted on page 34.

ASTROM, K. J.; WITTENMARK, Dr. B. **Adaptive Control**. 2nd. ed. New York: Dover Publications, inc, 2008. quoted 2 time on pages 11 and 17.

BLONDIN, J. Particle swarm optimization: A tutorial. **from site: http://cs. armstrong. edu/saad/csci8100/pso tutorial. pdf**, 2009. Quoted on page 31.

BORGES, F. G. et al. **Gaussian adaptive PID control with robust parameters considering plant parameter variation with optimization based on bioinspired metaheuristics**. 2019. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2019. quoted 3 time on pages 11, 18, and 22.

DEB, Kalyanmoy; KUMAR, Amarendra. Real-coded genetic algorithms with simulated binary crossover: Studies on multimodal and multiobjective problems. **Complex Systems**, v. 9, 1995. Quoted on page 26.

EIBEN, A.E.; SMITH, J.E. **Introduction to Evolutionary Computing**. [S.l.]: Springer Berlin Heidelberg, 2015. Quoted on page 23.

FERREIRA, Diógenes V.M.; COELHO, Vitor N. Calibration of a type 2 controller applied to a buck converter: a multi-objective analysis. **Electronic Notes in Discrete Mathematics**, Elsevier BV, v. 66,

p. 199–206, abr. 2018. Disponível em: <https://doi.org/10.1016/j.endm.2018.03.026>. Quoted on page 22.

GOLDBERG, David E. **Genetic Algorithms in Search, Optimization and Machine Learning**. 1st. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201157675. Quoted on page 26.

____. Genetic and evolutionary algorithms come of age. **Commun. ACM**, ACM, New York, NY, USA, v. 37, n. 3, p. 113–119, mar. 1994. ISSN 0001-0782. Quoted on page 25.

HAIDEKKER, Mark A. 15 - the pid controller. In: HAIDEKKER, Mark A. (Ed.). **Linear Feedback Controls (Second Edition)**. Second edition. [S.l.]: Elsevier, 2020. p. 253 – 273. ISBN 978-0-12-818778-4. Quoted on page 16.

HOGG, R.V.; TANIS, E.A. **Probability and Statistical Inference**. [S.l.]: Pearson/Prentice Hall, 2010. (Probability and Statistical Inference, v. 978, nos. 0-58475). ISBN 9780321584755. Quoted on page 35.

HOLLAND, J.H. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. [S.l.]: University of Michigan Press, 1975. ISBN 9780472084609. quoted 3 time on pages 22, 25, and 26.

HONG, Tzung-Pei; WANG, Hong-Shung. A dynamic mutation genetic algorithm. In: **1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No.96CH35929)**. [S.l.: s.n.], 1996. v. 3, p. 2000–2005 vol.3. ISSN 1062-922X. Quoted on page 26.

HUANG, S.N.; TAN, K.K.; LEE, T.H. A combined pid/adaptive controller for a class of nonlinear systems. **Automatica**, v. 37, n. 4, p. 611 – 618, 2001. ISSN 0005-1098. Quoted on page 16.

JALILVAND, A; KIMIYAGHALAM, A; ASHOURI, A; KORD, H. Optimal tuning of pid controller parameters on a dc motor based on advanced particle swarm optimization algorithm. **optimization**, Citeseer, v. 11, p. 12, 2011. Quoted on page 22.

Jinghui Zhong; Xiaomin Hu; Jun Zhang; Min Gu. Comparison of performance between different selection strategies on simple genetic algorithms. In: **International Conference on Computational**

Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06). [S.l.: s.n.], 2005. v. 2, p. 1115–1121. quoted 2 time on pages 23 and 24.

KASTER, Mauricio; BORGES, Fabio; FILHO, Marco; SIQUEIRA, Hugo; CORRêA, Fernanda. Comparison of several genetic algorithm strategies on a nonlinear gapid controller optimization applied to a buck converter. In: **Congresso Brasileiro de Automática (CBA 2018)**. [S.l.: s.n.], 2018. Quoted on page 22.

KASTER, M. S.; CUNHA, C.; BURGARDT, I.; NASCIMENTO, C. Análise do uso de controle PID não linear aplicado a conversor buck. **Proceedings of the X Conferência Brasileira de Dinâmica, Controle e Aplicações.**, 2011. quoted 3 time on pages 11, 17, and 19.

KASTER, Mauricio S.; PAGANO, Daniel J. Self-sustained oscillations generation in single-phase inverters. **Control & Automation Magazine (Revista Controle & Automação)**, v. 17, p. 60–69, January 2006. (in portuguese). Quoted on page 11.

KENNEDY, James; EBERHART, Russell C.; SHI, Yuhui. chapter seven - the particle swarm. In: KENNEDY, James; EBERHART, Russell C.; SHI, Yuhui (Ed.). **Swarm Intelligence**. San Francisco: Morgan Kaufmann, 2001, (The Morgan Kaufmann Series in Artificial Intelligence). p. 287 – 325. ISBN 978-1-55860-595-4. Quoted on page 30.

KENNEDY, R.; J., Eberhart. Particle swarm optimization. In: **Proceedings of IEEE International Conference on Neural Networks IV, pages**. [S.l.: s.n.], 1995. v. 1000. Quoted on page 30.

KRUSKAL, William H.; WALLIS, W. Allen. Use of ranks in one-criterion variance analysis. **Journal of the American Statistical Association**, Informa UK Limited, v. 47, n. 260, p. 583–621, dez. 1952. Disponível em: <https://doi.org/10.1080/01621459.1952.10483441>. Quoted on page 42.

LIN, Chuan; QING, Anyong; FENG, Quanyuan. A comparative study of crossover in differential evolution. **Journal of Heuristics**, v. 17, n. 6, p. 675–703, Dec 2011. ISSN 1572-9397. Quoted on page 29.

LINDFIELD, George; PENNY, John. Chapter 9 - optimization methods. In: LINDFIELD, George; PENNY, John (Ed.). **Numerical Methods (Fourth Edition)**. Fourth edition. [S.l.]: Academic Press, 2019. p. 433 – 483. ISBN 978-0-12-812256-3. Quoted on page 21.

MANIAS, Stefanos N. 7 - dc–dc converters. In: MANIAS, Stefanos N. (Ed.). **Power Electronics and Motor Drive Systems**. [S.l.]: Academic Press, 2017. p. 501 – 611. ISBN 978-0-12-811798-9. Quoted on page 14.

MARGHANY, Maged. Principles of genetic algorithm. In: **Synthetic Aperture Radar Imaging Mechanism for Oil Spills**. [S.l.]: Elsevier, 2020. p. 169–185. Quoted on page 25.

MENDEZ, Efrain et al. Novel design methodology for DC-DC converters applying metaheuristic optimization for inductance selection. **Applied Sciences**, MDPI AG, v. 10, n. 12, p. 4377, jun. 2020. Disponível em: <https://doi.org/10.3390/app10124377>. Quoted on page 22.

MICHALEWICZ, Zbigniew. Evolution strategies and other methods. In: ____. **Genetic Algorithms + Data Structures = Evolution Programs**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. p. 159–177. ISBN 978-3-662-03315-9. quoted 2 time on pages 23 and 25.

MILLER, B. L.; GOLDBERG, D. E. et al. Genetic algorithms, tournament selection, and the effects of noise. **Complex systems**, [Champaign, IL, USA: Complex Systems Publications, Inc., c1987-, v. 9, n. 3, p. 193–212, 1995. Quoted on page 25.

OGATA, Katsuhiko. **Modern Control Engineering**. 4th. ed. USA: Prentice Hall PTR, 2001. ISBN 0130609072. quoted 2 time on pages 11 and 16.

PENCHEVA, Tania; ATANASSOV, Krassimir; SHANNON, A. Modelling of a stochastic universal sampling selection operator in genetic algorithms using generalized nets. p. 1–7, 01 2010. Quoted on page 24.

PUCHTA, Erickson; LUCAS, Ricardo; FERREIRA, Felipe; SIQUEIRA, Hugo; KASTER, Mauricio. Gaussian adaptive pid control optimized via genetic algorithm applied to a step-down dc-dc converter. In: . [S.l.: s.n.], 2016. p. 1–6. quoted 3 time on pages 11, 22, and 31.

Puchta, E. D. P.; Siqueira, H. V.; Kaster, M. d. S. Optimization tools based on metaheuristics for performance enhancement in a gaussian adaptive pid controller. **IEEE Transactions on Cybernetics**, v. 50, n. 3, p. 1185–1194, 2020. quoted 2 time on pages 18 and 19.

ROWLEY, C. W.; BATTEN, B. A. Dynamic and closed-loop control. **Fundamentals and Applications of Modern Flow Control (Progress in Astronautics and Aeronautics, Vol. 231), American Institute of Aeronautics and Astronautics, Reston, VA**, p. 115–148, 2008. Quoted on page 15.

SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). **Biometrika**, Oxford University Press (OUP), v. 52, n. 3-4, p. 591–611, dez. 1965. Quoted on page 40.

STORN, Rainer; PRICE, Kenneth. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341–359, Dec 1997. ISSN 1573-2916. Quoted on page 27.

TAKAHASHI, Ricardo H. C.; PERES, Pedro L. D.; BARBOSA, Leandro L. S. A sliding mode controlled sinusoidal voltage source with ellipsoidal switching surface. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, v. 46, n. 6, p. 714–721, June 1999. Quoted on page 11.

Ting, C.; Ko, C. Incorporating tabu search into the survivor selection of genetic algorithm. In: **2008 IEEE International Conference on Systems, Man and Cybernetics**. [S.l.: s.n.], 2008. p. 553–558. Quoted on page 25.

TUSAR, Tea; FILIPIC, Bogdan. Differential evolution versus genetic algorithms in multiobjective optimization. v. 4403, p. 257–271, 05 2007. quoted 2 time on pages 27 and 37.

TUSSET, A. M.; BUENO, A. M.; NASCIMENTO, C. B.; KASTER, M. S.; BALTHAZAR, J. M. Nonlinear state estimation and control for chaos suppression in mems resonator. **Shock and Vibration**, v. 20, p. 749–761, 2013. Quoted on page 11.

WOLPERT, David; MACREADY, William. Macready, w.g.: No free lunch theorems for optimization. ieee transactions on evolutionary computation 1(1), 67-82. **Evolutionary Computation, IEEE Transactions on**, v. 1, p. 67 – 82, 05 1997. Quoted on page 21.

XU, Y.; HOLLERBACH, J.M.; MA, D. A nonlinear pd controller for force and contact transient control. **IEEE Control Systems**, v. 15, n. 1, p. 15–21, 1995. Cited By 91. Quoted on page 16.

YANG, Xin-She. Chapter 1 - introduction to algorithms. In: YANG, Xin-She (Ed.). **Nature-Inspired Optimization Algorithms**. Oxford: Elsevier, 2014. p. 1 – 21. ISBN 978-0-12-416743-8. Quoted on page 21.

ZITEK, Pavel; FIšER, Jaromír; VYHLíDAL, Tomáš. Iae optimization of pid control loop with delay in pole assignment space. **IFAC-PapersOnLine**, v. 49, n. 10, p. 177 – 181, 2016. ISSN 2405-8963. 13th IFAC Workshop on Time Delay Systems TDS 2016. Quoted on page 34.

APPENDIX A – SHAPIRO-WILKS TEST RESULTS

Table 8 – Shapiro-Wilks test results for GA

PSO Variation	P-Value	null hypothesis
GA 1	0.3124	0
GA 2	0.0341	1
GA 3	0.0100	1
GA 4	0.0003	1
GA 5	0.1056	0
GA 6	0.0022	1
GA 7	0.2515	0
GA 8	0.0001	1
GA 9	0.0577	0
GA 10	0.9962	0

Table 9 – Shapiro-Wilks test results for PSO

PSO Variation	P-Value	null hypothesis
PSO 1	0.5330	0
PSO 2	0.0357	1
PSO 3	0.1113	0
PSO 4	0.6388	0
PSO 5	0.0838	0
PSO 6	0.7055	0
PSO 7	0.3818	0
PSO 8	0.0729	0
PSO 9	0.4756	0
PSO 10	0.3828	0

Table 10 – Shapiro-Wilks test results for DE

PSO Variation	P-Value	null hypothesis
DE 1	0.0000	1
DE 2	0.0001	1
DE 3	0.0754	0
DE 4	0.4394	0
DE 5	0.0008	1
DE 6	0.0070	1
DE 7	0.4101	0
DE 8	0.4819	0
DE 9	0.1726	0
DE 10	0.3508	0