FEDERAL UNIVERSITY OF TECHNOLOGY – PARANA

RESEARCH AND GRADUATE DIRECTORATE

MASTER'S DEGREE IN ELECTRICAL ENGINEERING

FABIO GALVÃO BORGES

GAUSSIAN ADAPTIVE PID CONTROL WITH ROBUST PARAMETERS CONSIDERING PLANT PARAMETER VARIATION WITH OPTIMIZATION BASED ON BIOINSPIRED METAHEURISTICS

DISSERTATION

PONTA GROSSA

2019

FABIO GALVÃO BORGES

GAUSSIAN ADAPTIVE PID CONTROL WITH ROBUST PARAMETERS CONSIDERING PLANT PARAMETER VARIATION WITH OPTIMIZATION BASED ON BIOINSPIRED METAHEURISTICS

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

Advisor: Prof. Dr. Maurício dos Santos Kaster

PONTA GROSSA

2019

Ficha catalográfica elaborada pelo Departamento de Biblioteca
da Universidade Tecnológica Federal do Paraná, Campus Ponta Grossa
n.80/19

B732 Borges, Fabio Galvão

Gaussian adaptive PID control with robust parameters considering plant parameter variation with optimization based on bioinspired metaheuristics. / Fabio Galvão Borges, 2019.
75 f. : il. ; 30 cm.

Orientador: Prof. Dr. Maurício dos Santos Kaster

Dissertação (Mestrado em Engenharia Elétrica) - Programa de Pós-Graduação em Engenharia Elétrica. Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2019.

1. Sistemas de controle ajustável. 2. Algorítmos genéticos. 3. Partículas (Física, química, etc.). I. Kaster, Maurício dos Santos. II. Universidade Tecnológica Federal do Paraná. III. Título.

CDD 621.3

Elson Heraldo Ribeiro Junior. CRB-9/1413. 18/12/2019.

Federal University of Technology – Parana Câmpus Ponta Grossa
Board of Research and Graduate Studies **GRADUATE PROGRAM IN ELECTRICAL ENGINEERING**

APPROVAL SHEET

Title of Dissertation N^o 00/2019

GAUSSIAN ADAPTIVE PID CONTROL WITH ROBUST PARAMETERS CONSIDERING PLANT PARAMETER VARIATION WITH OPTIMIZATION BASED ON BIOINSPIRED METAHEURISTICS

by

FÁBIO GALVÃO BORGES

This Dissertation was presented at 9:00AM, May 30th, 2019 as a partial requisite to obtain the Master's Degree in Electrical Engineering, in the area of concentration *Control and Instrumentation* and in the research field of *Instrumentation and Control*, of the Graduate Program in Electrical Engineering. The candidate has been argued by the Examination Board comprised by following professors. After deliberation, the Examination Board considered the work approved.

Prof. Dr. Maurício dos Santos Kaster
Advisor

Prof. Dr. Grace Silva Deaecto
Janzen UNICAMP

Prof. Dr. Frederic Conrad
UTFPR

Prof. Dr. Fernanda Cristina Corrêa
UTFPR

Prof. Dr. Angelo Marcelo Tusset
Coordinator of PPGEE

The signed Approval Sheet are at the
Academic Record Department of UTFPR – Câmpus Ponta Grossa

ACKNOWLEDGEMENTS

This work would not have been possible without the financial support of CAPES Foundation to which I would like to thank profoundly.

To my family for always supporting me.

To my advisor, Mauricio Kaster for guiding me all this way, for his patience, for staying awake until late just to help me and being more than an advisor, but also a friend.

To all teachers from the department, each contributed in a significant way. A special thanks to professors Fernanda, Frederic, Hugo and Stevan who spent the most time helping me develop the best work possible.

To all my colleagues, especially my fellow friends from LICON.

To former and recent coordinators Claudinor and Angelo for their guidance, kindness and support.

ABSTRACT

BORGES, Fabio Galvão. **Gaussian Adaptive PID Control with Robust Parameters Considering Plant Parameter Variation With Optimization Based on Bioinspired Metaheuristics**. 2019. 75 p. Dissertation (Master's Degree in Electrical Engineering) – Federal University of Technology – Parana. Ponta Grossa, 2019.

The purpose of this work is to compare a linear PID to the Gaussian Adaptive PID control (GAPID) regarding their robustness to changes and variation on two different plants. The first one is the second order plant DC-DC Buck converter used as a study plant an analyzed through simulation. The second plant is a DC motor with a beam attached to it. An experimental prototype was built for this second plant to test the GAPID in a a real experiment. The Gaussian function has as adjustment parameters its convavity and the lower and upper bound of the gains. It is a smooth function with smooth derivatives. As a result, it helps avoid problems related to abrupt gains transition, commonly found in other adaptive methods. Because there is no mathematical methodology to set these parameters, two bio-inspired optimization algorithms were used, the Genetic Algorithm (GA) and the Particle Swarm Optimization (PSO). Functions to evaluate the results, called fitness functions, are necessary for the algorithms and were also used as performance comparison. A new variation to the fitness is proposed and the results demonstrate an improvement regarding the overshoot. Results also prove the robustness of the GAPID compared to the linear PID by load and gain sweep tests, achieving fast response (low settling time) and minimal variation, which is not possible to achieve when using the linear PID.

Keywords: Adaptive Control. GAPID. Genetic Algorithm. PSO.

RESUMO

BORGES, Fabio Galvão. **Controle PID Adaptativo Gaussiano com Parâmetros Robustos Considerando Variação dos Parâmetros da Planta com Otimização Baseada em Metaheurísticas Bioinspiradas**. 2019. 67 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2019.

O objetivo deste trabalho é comparar um PID linear ao controle PID Adaptativo Gaussiano (GAPID) quanto à sua robustez a mudanças e variações em duas plantas diferentes. A primeira é o conversor DC-DC Buck de segunda ordem utilizado como planta de estudo e analisado por simulação. A segunda planta é um motor de corrente contínua com uma viga ligada a ele. Um protótipo experimental foi construído para esta segunda planta para testar o GAPID em um experimento real. A função gaussiana tem ganhos de limite inferior e superior e concavidade como parâmetros. É uma função suave com derivadas suaves. Como resultado, ajuda a evitar problemas relacionados à transição abrupta de ganhos, comumente encontrados em outros métodos adaptativos. Como não há metodologia matemática para definir esses parâmetros, foram utilizados dois algoritmos de otimização bio-inspirados, o Algoritmo Genético (GA) e o por Enxame de Partículas (PSO). Funções para avaliar os resultados, chamadas de funções de adequação (fitness), são necessárias para os algoritmos e também foram usadas como comparação de desempenho. Uma nova variação para a função fitness é proposta e os resultados provam uma melhoria em relação ao overshoot. Os resultados também comprovam a robustez do GAPID em relação ao PID linear por testes de varredura de carga e ganho, obtendo resposta rápida (baixo tempo de estabilização) e variação mínima, o que não é possível atingir usando o PID linear.

Palavras-chave: Controle adaptativo. GAPID. Algoritmo Genético. PSO.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

BUCK CONVERTER SYMBOLS

L	Inductance	[H]
C	Capacitance	[F]
R	Load resistance	[Ω]
i_L	Inductor current	[H]
v_c	Capacitor voltage	[V]
v_o	Output voltage	[V]
V_i	Input voltage (power supply)	[V]
$d(t)$	Discrete PWM signal	
$u(t)$	Control signal	

GAUSSIAN ADAPTIVE PID SYMBOLS

k_0	Limit of Gaussian curve for error$= 0$
k_1	Limit of Gaussian curve for error$\to \infty$
q	Gaussian concavity factor
σ	Error

GENETIC ALGORITHM SYMBOLS

IAE	Integral Absolute Error
Fit	Fitness value
p_i	probability of the i-th individual

PARTICLE SWARM OPTIMIZATION SYMBOLS

v	Velocity of particle
v_i	Velocity of particle in previous iteration
r_1	Neighborhood Random supply number
c_1	Neighborhood Personal acceleration factor
r_2	Global Random supply number
c_2	Global acceleration factor

x	Particle position
x_i	Particle position in previous iteration
p_{best}	Particle Best position
g_{best}	Global Best position of all particles
w	acceleration factor
O_s	Overshoot

CONTENTS

1 INTRODUCTION

Most engineers and scientists are being acquainted with automatic control due to its importance and even being intrinsic to many systems, such as space vehicles, robotics, manufacturing, and most system that require control of temperature, pressure, position, flow etc. Automatic control systems introduced some advantages, for instance, response time and precision. More than half of the automatic controllers in industry use the Proportional Integral Derivative (PID) technique or modifications of it, because of its simplicity, well known design procedures and only three parameters to be tuned, working well in most systems and becoming well established (OGATA, 2010; NISE, 2007).

Despite working well on most systems, the PID controller has its limitations, showing poor performance when controlling plants with resonances, integrators, long time delay or unstable transfer functions. It also has no strategies regarding changes on the plant. Although still being able to control the system, it is not optimal and it may decrease performance significantly depending on these changes. As a result, significant interest arised from industry to improve the robustness of PID controllers. Adaptive controllers emerged as possible solutions (SUNG; LEE, 1996; ATHERTON D. P ; MAJHI, 1999).

There was extensive adaptive control research during the 1950s to design autopilots for high-performance aircraft. Constant gain linear feedback control could work well on one operating condition but not the entire flight. Therefore, a controller that could work on a wide range of operating conditions was needed. According to Åström K. J. ; Wittenmark (2013), "an adaptive controller is a controller with adjustable parameters and a mechanism for adjusting the parameters" (ÅSTRÖM K. J. ; WITTENMARK, 2013).

Some authors proposed methods of adaptive control using a set of constant gains that change according to some operating conditions (ALEXANDROV, 1999; SINGH; KUMAR, 2015; WANG; ZHANG, 2016). Albeit simple and efficient, those techniques rely on gain transitions that could be abrupt and cause unexpected and undesirable effects. Some others used adaptive continuous functions (XIAO et al., 2010; JIMENEZ-URIBE et al., 2015).

Because of its wide employment, some companies refuse the possibility to replace their PID control system to a more complex and costly one. For this reason, some authors proposed adaptive strategies linked to the PID. Rifai (2009) developed a methodology for Lyapunov-based adaptive PID control for different non linearly-parameterized series and parallel PID. Xiong e Fan (2007) proposed an algorithm that combines PID control scheme and model reference adaptive control (MRAC) to self tune the PID controller parameters in real time when system performances change and applied to control a DC electromotor drive.

Kaster et al. (2011) proposed the use of Gaussian functions applied to a PID to improve the control of a DC-DC Buck converter. Later on, Puchta et al. (2016b) further improved the Gaussian adaptive control by using bio-inspired optimization algorithms to find the Gaussian function parameters. In this work, no parameter variation of the plant was considered.

Nature-inspired algorithms (NIAs), as the name suggests, are algorithms whose source of inspiration is nature. NIAs can further be classified into algorithms based on Swarm Intelligence (SI), like Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Biological Phenomena. The other classification is called Bio-inspired (BI), like Genetic Algorithms (GA) and Simulated Annealing (SA). SI based algorithms are called intelligent because they are known to learn and improve their performance by observing the output on previous moves made by them. NIAs provide an efficient solution to many real-world optimization problems which are categorized to be hard or complex problems. The GA usual areas of application are Resource Scheduling, Airlines Revenue Management, Artificial Creativity, Bioinformatics, Clustering, Multi-modal Optimization and Multidimensional Systems. The PSO is mostly applied to Resource scheduling, Travelling Salesman Problem (TSP), Clustering methods, network generation methods, multidimensional and industrial optimization methods (KAPUR, 2015; JR et al., 2013).

Swiech, Oroski e Arruda used Genetic Algorithms (GA) to tune decoupled Proportional Integral Derivative (PID) controllers for distillation columns, instead of the classic tuning methods largely applied in industry like Ziegler Nichols and Biggest Log Modulus. In their work, two PID controllers were responsible for manipulating the re-flux mass flow rate and the flow of steam to control the composition of the bottom and top products. So the Genetic Algorithm found 6 optimal values of gains K_p, K_i and K_d for each controller. Although not an adaptive control, a bio-inspired algorithm like the Genetic Algorithm was applied to tune two linear PIDs and results proved it was efficient in achieving solutions that controlled well the output products.

The main advantage of the GA as a tuning method is that the two controllers could be tuned in a unified way to solve a multiobjective problem, which can be a challenging task to engineers in industry (SWIECH; OROSKI; ARRUDA, 2005).

Despite previous works explored the GAPID as an adaptive method to improve the PID controller, none analyzed the behavior of the controller when parameters of the plant are changed or created a method to guarantee the adaptive PID to maintain optimal performance even with those changes.

This work proposes a robust Gaussian Adaptive PID control, applied to control the output voltage of a DC-DC Buck converter, with robust parameters found and optimized by two nature-inspired algorithms, the GA and the PSO, by simulating variations on the Buck. This new control method ought to maintain close to optimal performance despite the plant variations.

The Buck converter is a relatively simple converter present on most low voltage power supplies, and represents a typical second-order plant. Its model is straightforwardly obtained by employing Kirchoff laws. So, it represents a good model to test the proposed control strategy without unpredicted model issues. It must be clearly stated that such model was used on simulations only. But it is advisable to test the control strategy in a different plant, preferably in real-world conditions. Then, a motor with distinct beams attached to it was used, and a National-based controller programmed in Labview language, very common in industry, was utilized. As every physical prototype, inherent noises and component imperfections arise. The proposed control technique must survive and give the expected responses in order to demonstrate its effectiveness.This is the case of the proposed GAPID, which results are presented in the following chapters.

1.1 OBJECTIVES

1.1.1 General Objective

The main objective of this work is develop a Gaussian Adaptive PID control (GAPID) optimized by bio-inspired methauristic algorithms for a computer simulation model of a second order study plant DC-DC Buck converter and for a real prototype plant of a DC motor using a PLC (Programmable Logic Controller to control the rotation speed and test its performance by applying load variation.

1.1.2 Specific Objectives

- Extract the mathematical model of the DC-DC Buck converter
- Develop a computation model of the converter to be used in simulations
- Build the GA and PSO algorithms for the application
- Run and test the algorithms with a load sweep variation and compare the results to demonstrate the GAPID adaptation on the plant
- Compare their performance to the linear PID

- Develop a simulation model to the DC motor plant
- Choose the best algorithm between the GA and PSO to develop the GAPID.
- Run and test the optimized GAPID with a load sweep for the DC motor by using metal beams of different mass.
- Apply the GAPID on the real prototype plant and compare to a linear PID

1.2 STRUCTURE

This work is structured as follows: chapter 2 describes three aspects: (i) the application circuit, a DC-DC buck converter, its mathematical model and transfer function; (ii) a brief discussion about the traditional PID control; (iii) and its nonlinear version: the Gaussian Adaptive PID control, which is the main control technique analyzed in this work. It also presents the optimization tools, based on bio-inspired metaheuristics, for finding the proper parameters for the GAPID controller. Two algorithms are employed: the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Some variants of the algorithms are also inspected, in order to compare their performances.

Chapter 3 presents the development of this work, how the simulation and experimental models were built, how the GAPID is applied to the system and how the algorithms find its parameters. In addition, this chapter presents which output results were captured, and how they were analyzed and compared.

Chapter 4 presents the obtained results. A comparison between the performance of the GA and PSO to optimize teh GAPID. The results of a load sweep are presented, using the best algorithm chosen and how the GAPID improves over the PID. To better understand the GAPID enhancement over PID, an analysis about the control signal saturation is performed.

After that, the validation of the simulation model of the experimental plant, the DC motor, is presented by comparing their speed output behavior with three metal beams of different mass (load sweep). Then, the output speed controlled by a PID is presented for the 3 beams. And at last, the output when controlled by the GAPID tuned by the optimization algorithm for each of the beams. The PID and GAPID results are then compared.

Finally, Chapter 5 presents the conclusions, interpreting and analyzing the results and their effects.

2 CONTROL STRATEGY APPLICATION

2.1 INTRODUCTION

This chapter will present the circuit application, a Buck converter, a typical second-order plant, which is widely used in low voltage switched power supplies, its working principle, its mathematical model and transfer function. Some comments about the traditional PID controllers, that is much employed in industry and can be applied to stabilize the voltage output of the converter, are included, as well as a discussion about its performance limitations, that can be overcomed by applying nonlinear controllers instead. The Gaussian Adaptive PID (GAPID) is one such controller, that will also be presented. It is also presented the limit control by using the sliding mode control technique. Here, this technique is used as a reference for the best control possible. It is compared to the GAPID and the PID to show how close they are to the optimal control.

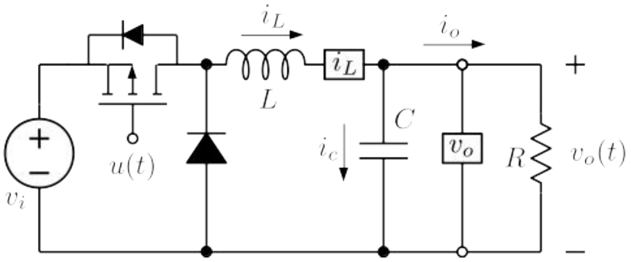
2.2 STUDY CIRCUIT PLANT: STEP-DOWN DC-DC CONVERTER

The buck converter is a DC-to-DC power converter which steps down voltage (while stepping up current) from its input (supply) to its output (load). It can be highly efficient, making it useful for applications such as the majority of low voltage power supplies used in electronic devices or as part of electric/electronic equipment like USB, DRAM and the CPU (1.8 V or less) (JULIAN et al., 1997).

The Buck converter was also used in some studies to cell phones and other portable applications. (XIAO et al., 2004; LIOU; YEH; KUO, 2008; MAITY et al., 2011). Rodriguez et al. (2010) made use of it for envelope tracking applications in radio frequency to improve the efficiency in wireless communication transmitters.

The step-down converter is presented in Fig. 1.

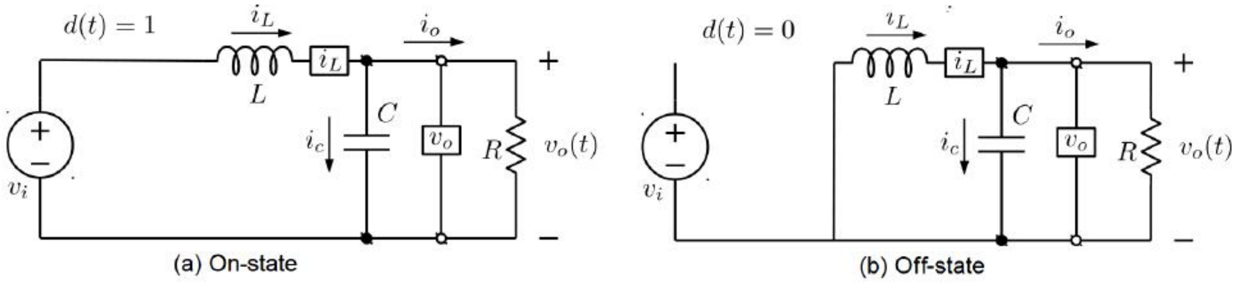
Graph 1 – Schematics of the Buck converter



source: own autorship

The converter operates in two states, one when the switch is on (Fig. **??**(a) and the other when its off (Fig. **??**(b)),

Figure 1 – Buck Converter States



(a) On-state

(b) Off-state

source: own autorship

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.

Through nodal analysis, it is possible to relate the inductor current with the current going through the capacitor and the load as in eq.(1):

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

Through mesh analysis, is possible to relate the input voltage to the voltages of the inductor and capacitor as in eq.(2):

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

From eq.(2), the input voltage on the inductor and capacitor are zero if the switch is off ($d = 0$). 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 Eq.s (3) and (4).

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

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

The step-down converter is a linear second-order, single input, single output system. So, a possible useful analysis tool is the Laplace transform (ROWLEY; BATTEN, 2008). Hence the control input-to-voltage output transfer function is written as in eq.(5).

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

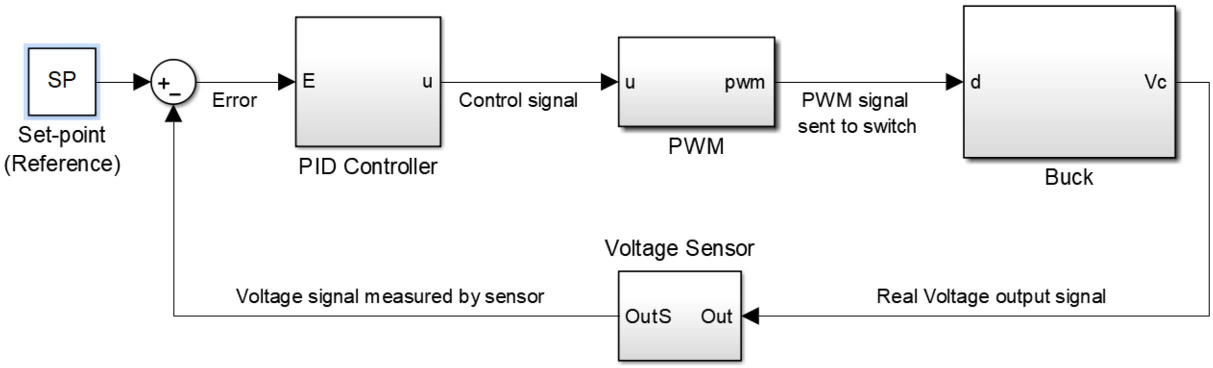
To assure the converter to work properly, a closed loop control is necessary to stabilize the output voltage to a desired value, which is the main purpose of the converter: to serve as a low-voltage power supply.

The most commonly used type of controller is the Proportional-Integral-Derivative (PID) controller, reaching around 95% of usage as control mesh in Industries all over the world. Even if new technology advances were made in new control techniques, the PID is still largely used in many industrial processes for its simplicity, easy application and performance (ÅSTRÖM K. J. ; HÄGGLUND, 1995).

The goal of automatic controllers, like the PID, is to maintain the values of the controlled variables in a specific, or close enough, value called set-point by acting and changing the manipulated variables (CORNETTI, 2014). In case of the Buck converter, one of the controlled variables is the output voltage and the manipulating variable is the opening and closing of the switch (changing the duty cycle). A typical closed-loop PID control diagram is shown in Fig. 2.

The output signal is measured by a sensor and then compared to the set-point or reference, this is the error analyzed by the PID controller so it can calculate whether the switch has to stay on longer or not. This on and off signal is sent to the switch through a pulse width modulator (PWM) generating the output response and closing the loop (ASTRÖM K. J. ; HÄGGLUND, 2004).

Figure 2 – PID closed-loop diagram



source: own autorship

The control signal $u(t)$ is the output sent by the controller and it is represented in eq.(6)

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

In eq.(6) the output control signal $u(t)$ depends on the error signal $e(t)$ which is the the difference between the set-point and the system output. $u(t)$ is also the sum of three parts composed of one proportional to the error signal, a second proportional to the integral of the error and a third proportional to the derivative of the error (CORNETTI, 2014). The proportional deal acts directly over the system gain, the integral deal is related to the precision of the output response and the derivative deal anticipates the error and tries to correct it in advance (SWIECH; OROSKI; ARRUDA, 2005).

Although widely used, the PID has its downsides just like other linear controllers. It has limited robustness and is sensitive to changes on the plant and the system in overall. For instance, the system may change over time, like some parts may get rusty or gears may get looser in a mechanical production line. Another disadvantage is that it is designed to function on specific loads. Taking a conveyor belt as an example, a heavier box may change the control response, pulling it out of an optimal operating condition. It could result in an energy waste or slowing down the belt.

One of the main disadvantages is the natural limitations of a linear controller: the gains are fixed and it poses performance limitations. It is known that a high proportional gain can speed-up the controller response but cause a higher overshoot, which is commonly an undesirable behavior in most systems. The derivative gain is also able to speed-up the controller response with lower overshoot but is very turns the controller very sensible to noise. In several cases, the derivative component is not employed due to this sensitivity.

Nonlinear controllers can overcome these limitations. An adaptive PID represents a modification of the traditional PID that can perform far better, achieving excellent performances with low overshoot and adapting the noise sensitivity as the controlled output approaches the setting-point. But the the design methodology of such controllers is much harder.

In section 2.4, an adaptive PID based on Gaussian functions (reference) is presented. This controller was proposed by (KASTER et al., 2016) and is the subject under study in this work. The linked parameters design methodology has the advantage of using the designed PID parameters into account and derive the Gaussian parameters by using an optimization tool.

2.3 EXPERIMENTAL PLANT: DC MOTOR

The experimental plant built for this thesis consists of controlling the rotation speed of a DC motor with a metal beam attached to its rotation axis shaft. The DC motor is an actuator that converts continuous electric current to rotating mechanical power, known as torque. There are several types of DC motors, and in this thesis, the DC motor type used is the fixed field or permanent magnet. This type of motor is often used because of its easy speed and angular position control. The control of these motors is achieved by determining suitably the voltage to be applied to the motor armature (DORF; BISHOP, 2001).

Considering that the magnetic field generated by the permanent magnet is fixed, it is possible to write the torque generated by the engine as shown in Eq. (7)

$$T = K_t . i_a \tag{7}$$

where:

$T =$ torque

$K_t =$ torque constant

$i_a =$ armature current

The counter electromotive force generated by the motor, can be described as proportional to the angular speed of the motor shaft $\theta^{'}$, as showed in Eq. (8).

$$e = K_b . \dot{\theta} \tag{8}$$

where:

$e =$ electromotive force

$K_b =$ electromotive force constant

$\dot{\theta} =$ angular speed

By applying the Newton's second law, the behavior of the motor mechanical motion is described by the following Eq. (9)

$$J\ddot{\theta} + b\dot{\theta} = K_t.i_a \tag{9}$$

where:

$J =$ the sum of the moments of inertia from the beam

$b =$ viscous friction from the moving parts of the motor

$K_t =$ torque constant

$i_a =$ armature current

$\dot{\theta} =$ angular speed

$\ddot{\theta} =$ angular acceleration

Applying the Kirchhoff's voltage law to the electric system, the behavior of the electric system of the motor is obtained as shown in Eq..

$$L\frac{di_a}{dt} + R.i_a = V_i - K_b.\dot{\theta} \tag{10}$$

where:

$L =$ armature inductance

$R =$ armature resistance

$V_i =$ input voltage

Eq.s (9) and (10) together represent the mathematical model of the DC motor. Both Eq.s are coupled through the electric current and angular speed.

With the beams attached, the system is influenced by the drag force, which is a force opposing the torque. Eq. (11) represents the drag force (JANZEN et al., 2014).

$$F_d = \rho.A.C_d.(\frac{V}{2})^2 \tag{11}$$

where:

$F_d =$ drag force

$\rho =$ air density

$C_d =$ drag coefficient

$V =$ linear speed

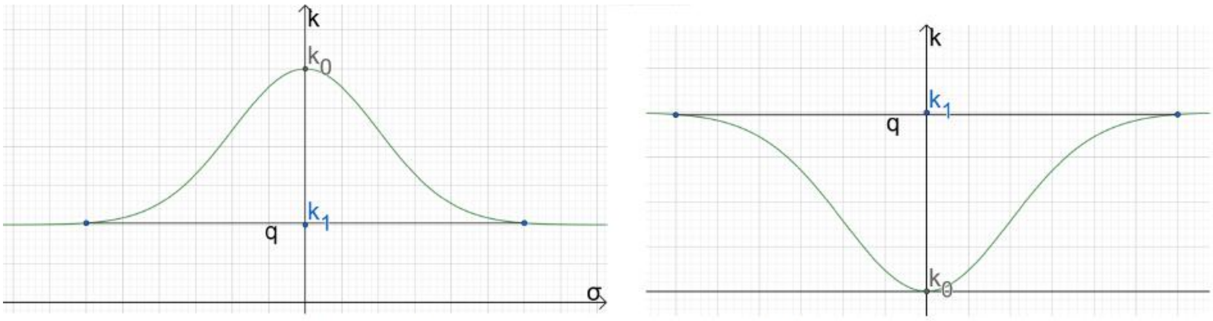
2.4 GAUSSIAN ADAPTIVE PID CONTROLLER

The Gaussian Adaptive PID controller (GAPID), is based on a linear PID controller, but takes into account variable gains instead of fixed gains, which gives to the GAPID the characteristic of adaptive. This controller considers Gaussian functions of the input error given by Eq. (12)

$$f(\sigma) = k_1 - (k_1 - k_0)e^{-q.\sigma^2} \tag{12}$$

Figure 3 demonstrates the behavior of an example of the Gaussian function

Figure 3 – Gaussian Function



(a) Downwards concavity Gaussian function

(b) Upwards concavity Gaussian function

source: own autorship

where:

σ is the error;

q defines the openness of the Gaussian function;

k_0 and k_1 are the bottom and upper limits. Depending on the error signal, the gain will vary between the limits defined by those values.

The Gaussian is a smooth function with smooth derivatives, making the gain change gradually as the error approaches zero, thus preventing abrupt changes that could damage the physical system. The parameter q affects this smooth transition, the bigger the value, the less opened is the function, turning the transition from the lower to upper limits steeper.

In Fig. (3) (a), the concavity is downwards, because $k_0 > k_1$. If $k_0 < k_1$, alternatively the concavity is upwards as in Fig.(3) (b).

Each branch of the controller, the proportional, the derivative and the integral, has its own Gaussian function with its own set of gains. That means there would be nine parameters: k_{p0}, k_{p1}, q_p, k_{i0}, k_{i1}, q_i, k_{d0}, k_{d1} and q_d. But to prevent noise issues there should be no derivative gain when the system reaches the steady state and the controllers acts as a PI, meaning that the Gaussian function for the derivative branch ought to be upwards ($k_{d1} > k_{d0}$) and $k_{d0} = 0$. Therefore the function for the derivative branch is simplified as shown in Eq. (13)

$$f_d(\sigma) = k_{d1}(1 - e^{-q_d.\sigma^2}) \tag{13}$$

consequently there are actually 8 parameters to be set.

The gaussian for the derivative gain is typically upwards and for the integral gain downwards. The proportional gaussian tends to be upwards, but it can be downwards in some cases. In the Buck converter, the function with concavity upwards is applied to the proportional and derivative branches, as the gain must be higher when the error is also high and drops as the error approaches zero. This assures a faster response time, and as the error drops, so does the proportional and derivative gains in order to prevent the signal to surpass the target value (overshoot). Whereas this should not be the case for the integral branch, where this could increase the settling time, delaying the response. For this reason low initial integral gain helps achieving a faster response, while high integral gain during the steady state assures faster convergence to a null error.

The 8 parameters to be set, the 5 gains and 3 q will be called free "parameters" from now on. There is another method to find those parameters. This alternate method makes use of the linear gains of the PID and links them to the GAPID's with a relation of $k_0 = x.k_{PID}$ $k_1 = \frac{1}{x}k_{PID}$. All relations are provided in Eq. (14)

$$\begin{aligned} &k_{p0} = x.kp_{PID}; \qquad k_{p1} = \tfrac{1}{x}.kp_{PID} \\ &k_{i0} = y.ki_{PID}; \qquad k_{i1} = \tfrac{1}{y}.ki_{PID} \\ &k_{d1} = z.kd_{PID} \end{aligned} \tag{14}$$

As a result this "linked parameters" method provides 6 parameters to be set: x, y, z, q_p, q_i, q_d, compared to 8 from the free parameters. This linked method has an advantage as it can use the same gains from an already tuned linear PID, thus enhancing its performance so it could just be added to an implemented controller in an industry, for example.

There is no mathematical method to obtain the appropriate values of these parameters yet. Optimization tools, like bio-inspired algorithms, are a good choice to find these parameters, and are explained in chapter 3.

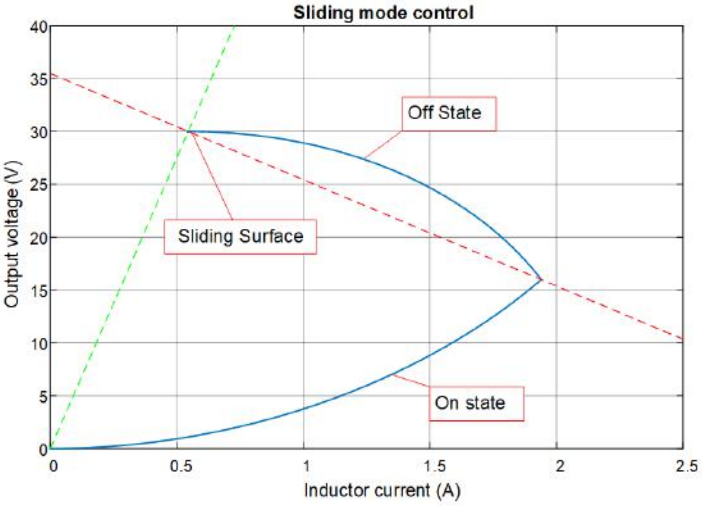
2.5 LIMIT CONTROL

The limit control refers to the best control possible for achieving the best performance regarding the controlled variables. Normally, the maximum and minimum levels of the control signal are used, in specific intervals chosen carefully in order to match the setpoint in the minimum time possible, so there is no way to perform better.

The limit control is used in this work as a reference to see how far other control strategies can approach the limit.

It can be derived from sliding mode control (SMC), a technique developed by Utkin (1978), where the designer can set a control law capable of making that all trajectories of this system converge to a determined surface called sliding surface. The engineer must choose this sliding surface to guarantee that all trajectories on it converge to the target operation value (set-point) (AGOSTINHO, 2009; OLIVEIRA, 2014). Fig. (4) shows in red the sliding surface for this work.

Figure 4 – Sliding mode control: Equilibrium points (green); sliding surface (red); trajectory of system states (blue).

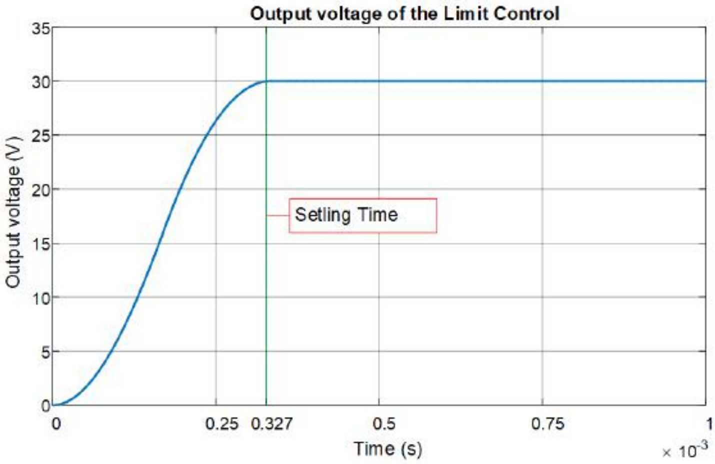


source: own autorship

An ideal switching surface can achieve global stability, good large-signal operation, and fast dynamics. After the trajectory reaches the inside of the sliding surface, it is said that the system is operating in sliding mode. When the system is in sliding mode, it is insensitive to parametric variations and external disturbances. This property ensures robustness to the SMC. However, due to the dependence of the output voltage ripple on the relative magnitudes of the input and output voltages, and load power, the sliding mode control is largely operated at a high variable switching frequency called chattering and also at high control gains. This will make the design of the filter components difficult in some applications (AGOSTINHO, 2009; YAN et al., 2009).

In Fig. (4) the steady state is reached considering only two switching states. It is an ON and OFF switching states with an exact duration of both that results in the fastest possible transient of the controlled variable (output voltage). It appears in the state-space by following the ON-state path, touching the switching surface and then following the OFF-state path that hits exactly the set-point. As a result, the sliding mode control is used here as the limit control (reference), where we can find the fastest possible startup transient for the controlled variable. Graph (2) shows the resulting output voltage.

Graph 2 – Output Voltage using SMC



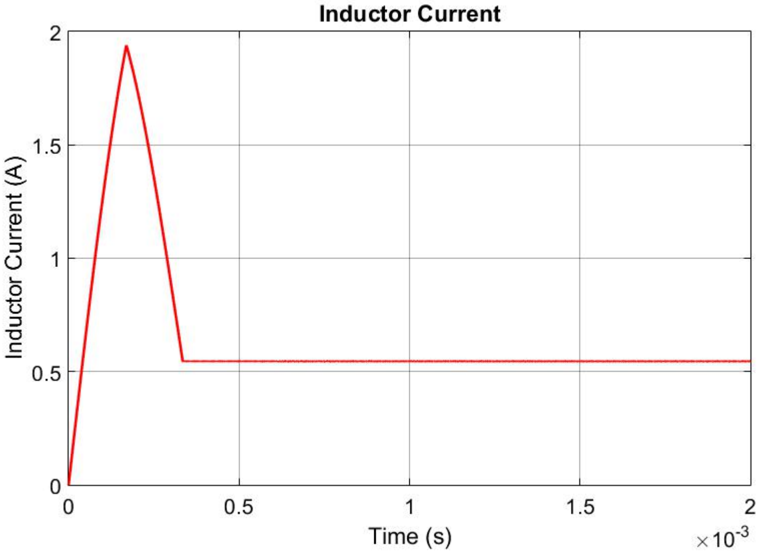
source: own autorship

The output voltage on Graph (2) represents the best performance it can reach, by the application of the limit control, with a minimum settling-time of 0.327 ms. This minimum setling-time will be used as a reference that indicates the maximum performance possible which is very useful to measure the margin still left for enhancement of any other controller solution, like the GAPID used in this work.

Figure 3 shows the corresponding inductor current.

There is a high peak inductor current during the transient period. Although undesirable in

Graph 3 – Inductor current



source: own autorship

real applications, it is not being considered because this work aims at the minimum settling-time of the controlled variable (output voltage).

2.6 PERFORMANCE EVALUATION METHOD

There must be an evaluation method to compare the different control strategies. In this case, a function must evaluate the performance of the controller by measuring the output voltage signal of the buck converter. Fig.(4) is an example of an output voltage signal.

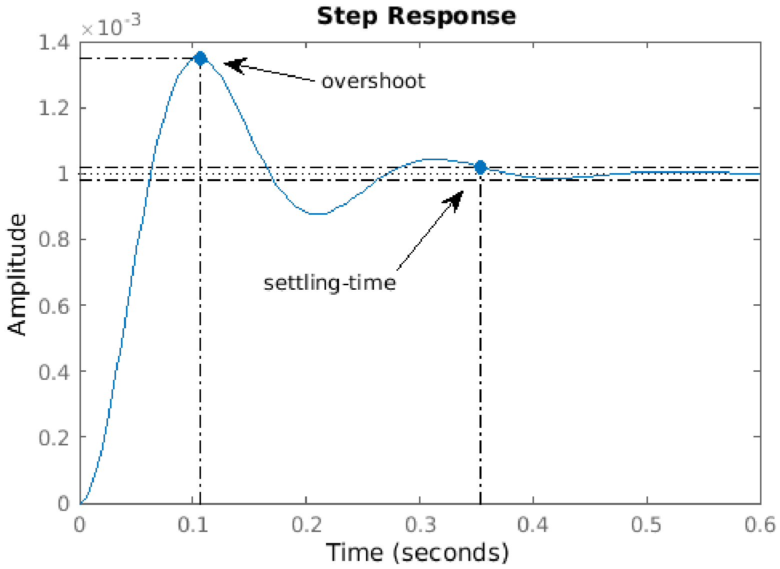
In figure 4 there is an example two common variables used to assess performance: overshoot and settling time of a step response. A controller is designed to minimize the settling time while containing the overshoot of the signal within acceptable limits, for this reason both values are commonly used as design requirements and can be used to compare controllers. The lower these two values, the better is the overall control (TAY; MAREELS; MOORE, 1997).

There are other methods that express performance as function of the error, which implicitly considers both setling-time and overshoot in a single function. The definition of the error signal is represented in eq. (15)

$$\sigma(t) = P - V_o(t) \tag{15}$$

where $\sigma(t)$ is the error, P is the set point value and $V_o(t)$ is the output voltage. Puchta et al. (2016b) compared four methods of evaluation and demonstrated that the Integral Absolute Error (IAE) function

Graph 4 – Output Signal Example



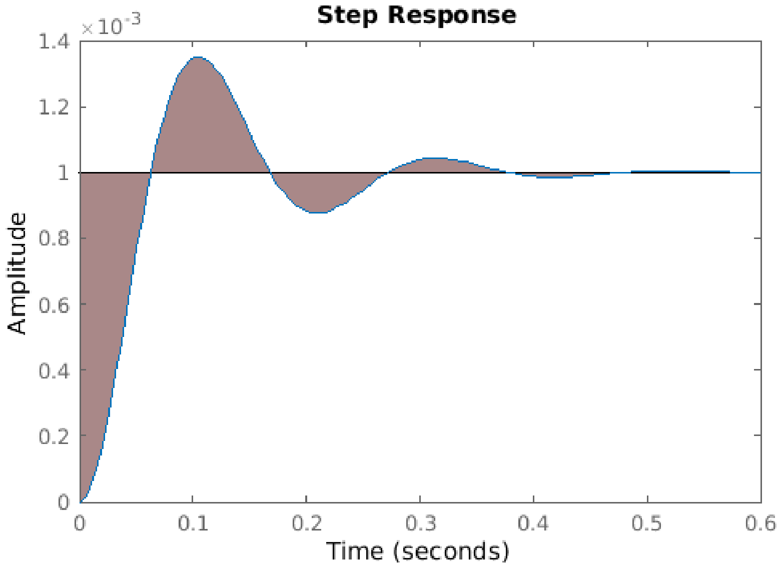
source: own autorship

was the one that presented the best consistent results. It is defined in eq. (16).

$$IAE = \int |\sigma(t)| dt \tag{16}$$

This function represents the area formed by the absolute value of the error signal, thus the output of this function is a positive number. An example can be seen in Fig.(5).

Graph 5 – Area corresponding to the IAE



source: own autorship

The lower the overshoot and the faster the output signal takes to reach the set point (Os and T_s), the smaller is the area, the better is the controller. Hence, the IAE function can be considered a cost function, that must be minimized. Instead of using the IAE function directly to be minimized by the algorithm, an alternate function will be used, called fitness function (Fit), that must be maximized. Eq. (17) represents the normalyzed Fitness function (assuming values ranging from 0 to 1).

$$Fit = \frac{1}{1 + IAE} \tag{17}$$

One problem with this equation is that it does not consider the overshoot and settling time separately and the algorithm could force a fast response but with high overshoots. An alternative is to change the Fitness equation applying a weighting (α) to the overshoot (O_s) in addition to the IAE. When the maximum voltage value surpasses the voltage target value SP, it is saved and added to IAE. Eq.(18) is the new fitness function.

$$\begin{aligned} O_s = & \quad V_{max} - SP \\ Fit = & \quad \frac{1}{1 + (IAE + \alpha O_s)} \end{aligned} \tag{18}$$

The fitness Eq. (eq.(18)) scales the fitness value between 0 and 1 and is the opposite from the $IAE + \alpha O_s$. On one hand, a low IAE and overshoot turns the fitness close to 1, on the other hand, a high IAE and overshoot turns the fitness close to 0. Consequently, higher values for the fitness function denote better performance.

2.7 METAHEURISTICS

2.7.1 Introduction

Where finding an optimal solution may be impossible or impractical, heuristic techniques can be used to speed up the process of finding a satisfactory one (IPPOLITI, 2014). Metaheuristic is defined, in computer science, as a high-level heuristic designed to find, generate or select a search algorithm that could provide some sufficient possible solutions (BIANCHI et al., 2009).

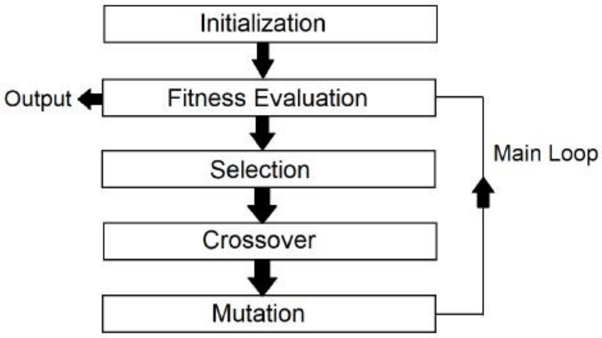
Genetic Algorithm(GA) and Particle Swarm Optimization(PSO) are both part of optimization search algorithms, used when the sample search space is too large for a traditional “blind” search (COLEY, 1999). In this work the two search algorithms strategies are used and compared to find possible optimal parameters of the GAPID, previously explained in section 2.4.

2.7.2 Genetic Algorithm

Genetic algorithm (GA) is a robust and efficient search technique that has been used to many engineering applications since it was introduced. The popularity of this method is based on simply solving multidimensional and multi-modal optimization problems without requiring any additional information such as the gradient of an objective function (CHEN; YIN, 2012).

GA was invented by John Holland, his students and colleagues at University of Michigan in the 1960s based on Darwin's theory of evolution. His original goal was to study the phenomenon of adaptation as it happens in nature and develop ways to import them to computer systems. Holland's GA consists on a method of moving from a generation of chromosomes to a new, more adapted one by applying a sort of "natural selection", crossover and mutation operations (MITCHELL, 1998). Fig. (5) shows the flowchart logic of implementing a GA.

Figure 5 – Flowchart of GA



source: own autorship

2.7.2.1 Initialization

In the first step a initial population is randomly generated with a set number of chromosomes. Each chromosome possess a group of genes, in case of this work these genes are the 8 parameters of the GAPID (k_{p0}, k_{p1}, q_p, k_{i0}, k_{i1}, q_i, k_{d1}, q_d), using the free parameters method, or 6 (x, y, z, q_p, q_i, d_d), using the linked parameters.

2.7.2.2 Fitness Evaluation

The goal of the GA is to optimize the values of the GAPID parameters in order to improve the controller. For that to happen, there must be a way to evaluate how "adapt" a chromosome is, or how good it is compared to the others. After the initial step, and every successive loop, each chromosome of the randomly generated population must be evaluated and given a rating. The evaluation is usually done by a function which should be minimized or maximized. In this case, the fitness function in Eq. (17) presented in section 2.6 is used by the genetic algorithm as a fitness evaluation.

2.7.2.3 Selection

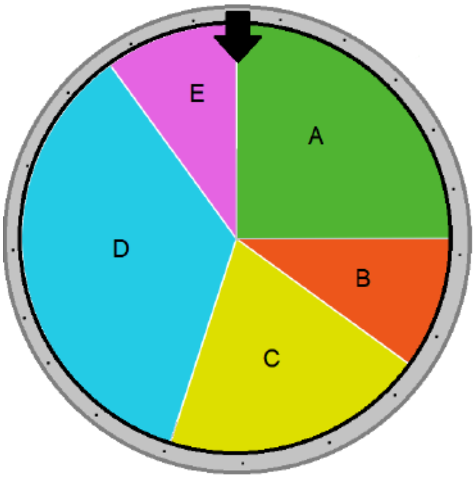
After the initial step and during each successive generation, a portion of the existing population is selected to breed a new generation. The selection is based on the fitness values, where fitter solutions are more likely to be selected (SCHMITT, 2001). The selection operation determines which solutions are to be preserved for reproduction and which ones are to die out during the iterated optimization process of GA (LU et al., 2013). Two selection methods were employed: the fitness proportionate selection, also known as roulette wheel selection, and the tournament selection.

In the fitness proportionate selection or roulette wheel, the fitness value is used to associate a probability of selection with each individual chromosome. Eq.(19) presents the probability of selection.

$$p_i = \frac{Fit_i}{\sum_{i=1}^{N} Fit_i} \tag{19}$$

where i is an individual chromosome, Fit_i is the fitness value of that chromosome and N is the number of chromosomes of the population (LIPOWSKI; LIPOWSKA, 2012). The probability can be compared to a roulette wheel, hence the name, like in Fig.(6).

Figure 6 – Roulette wheel



source: own autorship

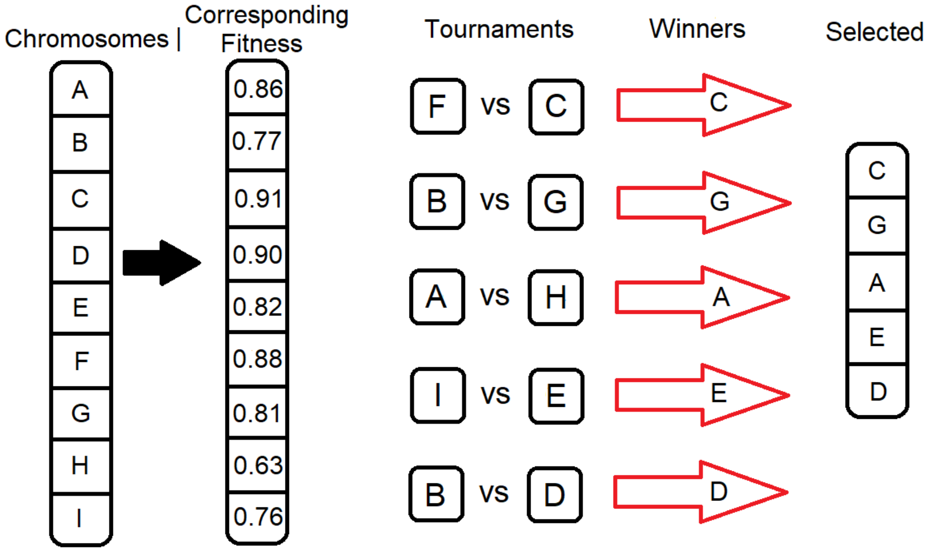
The roulette corresponds to the sum of probabilities. The fitter the chromosome, more space it fills in the roulette, consequently greater is its chance to be selected to pass on its genes to the next generation. Although the fittest chromosome is unlikely not to be selected and “die”, it can still occur without a method of elitism in the algorithm, which guarantees the fittest chromosome never dies (LIPOWSKI; LIPOWSKA, 2012).

The roulette spins and where the arrow is pointing when it stops indicates the selected chromosome (Random selection based on fitness). Taking fig.(6) just to illustrate, chromosome D has the highest fitness of the group, as a result there is a higher chance the roulette will stop pointing to its area. If it does so, D is selected for crossover and pass on its genes. Usually half the population is selected. For instance, in a population of 20 chromosomes, 10 will be selected based on the fitness to participate on the crossover. It is so because 10 chromosomes will breed a new population of 20, maintaining the population’s size.

It is important to point out that this process of higher probability of selection of the fitter chromosomes is what makes the algorithm converge faster to a more suitable solution, as they are more likely to pass on better genes to the next population. But as the process is dealing with probabilities it is still random and not so fit chromosomes can still pass on their genes. This chance is important to ensure variety to future generations and a chance to find other fitter solutions (GOLDBERG, 2002). That is also the goal of the mutation operation which will be explained further.

The second method of selection is the Tournament. It involves running several "tournaments" among some chromosomes chosen at random from the population. The winner of each tournament (the one with the better fitness between the two) is selected for crossover (MILLER; GOLDBERG et al., 1995).

Figure 7 – Example of Tournament



source: own autorship

In the example of fig(7) two random chromosomes are selected to compete on the first tournament and the fitter one is selected. Another two chromosomes are randomly selected to compete and once again the fitter one is selected. This process is repeated until half the population is selected for reproduction.

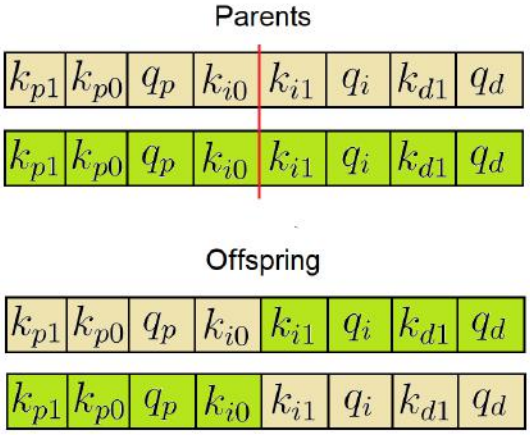
The tournament selection has a variation known as sudden death tournament. On the original tournament, a chromosome that “lost” goes back to the pool and can be randomly chosen to face another chromosome, getting a chance to be selected for the crossover. On the death tournament, however, the losing chromosome is immediately eliminated and can not be chosen to compete again. It works as an elimination tournament (MILLER; GOLDBERG et al., 1995).

2.7.2.4 Crossover

The crossover operator has been considered the main component of GAs and make them distinctively different from other optimization and problem solver algorithms (YAMADA; NAKANO, 1995). It selects groups of genes of a pair of parent chromosomes and mix then to create a new offspring, mimicking biological recombination between two single chromosome organisms (SHARMA; SINHA, 2014). In case of this work the genes are the parameters of the GAPID.

There are different methods of crossover and the simplest one is the single-point crossover. Fig.(8) illustrates this method (GWIAZDA, 2006).

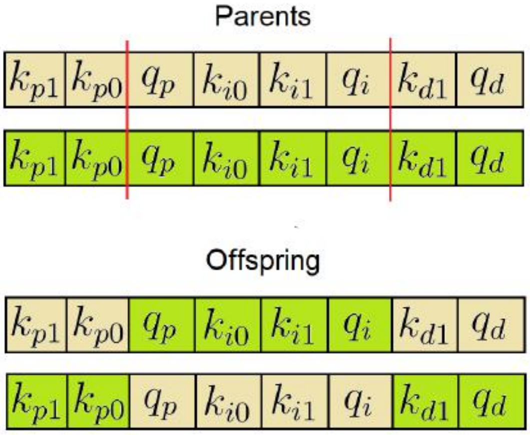
Figure 8 – Single-point crossover


Parents
k_{p1} k_{p0} q_p k_{i0} k_{i1} q_i k_{d1} q_d
k_{p1} k_{p0} q_p k_{i0} k_{i1} q_i k_{d1} q_d
Offspring
k_{p1} k_{p0} q_p k_{i0} k_{i1} q_i k_{d1} q_d
k_{p1} k_{p0} q_p k_{i0} k_{i1} q_i k_{d1} q_d

source: own autorship

In the two-point crossover, two points are picked randomly from the parents as in fig.(9)

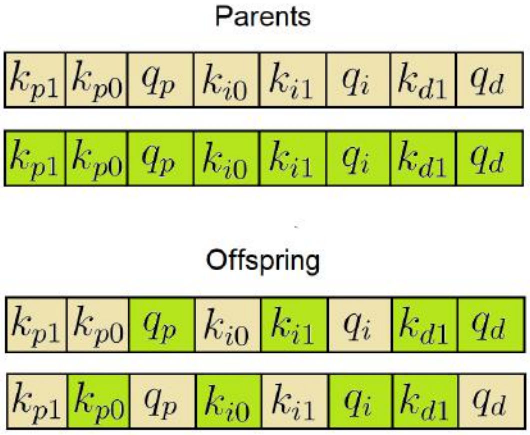
Figure 9 – Two-point crossover


Parents
k_{p1} k_{p0} q_p k_{i0} k_{i1} q_i k_{d1} q_d
k_{p1} k_{p0} q_p k_{i0} k_{i1} q_i k_{d1} q_d
Offspring
k_{p1} k_{p0} q_p k_{i0} k_{i1} q_i k_{d1} q_d
k_{p1} k_{p0} q_p k_{i0} k_{i1} q_i k_{d1} q_d

source: own autorship

There is also the uniform crossover exemplified in fig.(10), in which the single segments are exchange individually and randomly from the parents, so there is no bias in two close segments always coming from the same parent (GWIAZDA, 2006).

Figure 10 – Uniform crossover


Parents
k_{p1} k_{p0} q_p k_{i0} k_{i1} q_i k_{d1} q_d
k_{p1} k_{p0} q_p k_{i0} k_{i1} q_i k_{d1} q_d
Offspring
k_{p1} k_{p0} q_p k_{i0} k_{i1} q_i k_{d1} q_d
k_{p1} k_{p0} q_p k_{i0} k_{i1} q_i k_{d1} q_d

source: own autorship

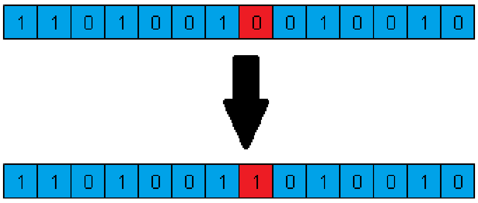
The crossover method influences how fast and accurate the algorithm will converge to optima solutions (GWIAZDA, 2006).

2.7.2.5 Mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to biological mutation.

Mutation alters one or more gene values in a chromosome from its initial state. In it, the solution may change entirely from the previous solution. Hence GA can come to a better solution. It occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set too high, the search will turn into a primitive random search. It can also prevent the algorithm to be stuck in a local optimal (PEZZELLA; MORGANTI; CIASCHETTI, 2008). Fig (11) shows a simple example of a mutation.

Figure 11 – Example of mutation



1 1 0 1 0 0 1 0 0 1 0 0 1 0

1 1 0 1 0 0 1 1 0 1 0 0 1 0

source: own autorship

2.7.3 Particle Swarm Optimization

The Particle Swarm Optimization (PSO) is a swarm-based computing technique developed by Kennedy and Eberhart, which was inspired in the social behavior of groups of animals, like flocks of birds and schools of fishes (KENNEDY, 1995). The hypothesis of sociobiologist E. 0. Wilson referencing fish schooling was fundamental to the development of their PSO algorithm.

In his hypothesis Wilson claims that, “in theory at least, individual members of the school can profit from the discoveries and previous experience of all other members of the school during the search for food. This advantage can become decisive, outweighing the disadvantages of competition for food items, whenever the resource is unpredictably distributed in patches.” (WILSON, 1975). He suggests that social sharing of information among members of the same specie offers an evolutionary advantage. So the main characteristic of the PSO is that simple agents, when working together, can arise a collective intelligence conduct, respecting simple rules.

Each particle presents a position and a velocity. The current position corresponds to a point in the search space, or a candidate solution. The initial swarm positions and velocities are random generated, limited by the search-space size. The initial fitness is calculated, setting the first personal position (p_{best}) and population global best (g_{best}), before starting the iterations. At each iteration, the particles’ velocity and position vectors are updated by Equations (20) and (21) respectively, based on the previous information of p_{best} and g_{best} (CLERC; KENNEDY, 2002).

$$v_i(t+1) = v_i(t) + r_1c_1(p_{best_i}(t) – x_i(t)) + r_2c_2(g_{best}(t) – x_i(t)) \tag{20}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{21}$$

where:

i = Index of the particle

$v_i(t)$ = Velocity of particle i at instant t;

r_1 and r_2 = Random supply numbers between 0 and 1;

c_1 = Personal acceleration factor;

c_2 = Social acceleration factor;

p_{best} = Best known personal position;

g_{best} = Best known global position;

$x_i(t)$ = Position of the particle i at instant t.

Each term of the velocity equation serves a purpose. The term $r_1c_1(p_{best_i}(t) - x_i(t))$ is called the cognitive component and acts as the particle's memory, causing it to tend to return to the regions of the search space in which it has experienced high individual fitness. The personal acceleration factor c_1 affects the size of the step the particle takes toward its individual best solution achieved so far. (BLONDIN, 2009)(SHI; EBERHART, 1998a)

The term $r_2c_2(g_{best}(t) – x_i(t))$, called the social component, causes the particle to move to the best region the whole swarm has found until that instant. The Social acceleration factor c_2 represents the size of the step the particle takes toward the global best candidate solution g_{best}. The random values r_1 and r_2 cause each particle to move in a semi-random manner heavily influenced in the directions of the personal best solution and global best solution of the swarm. (BLONDIN, 2009)(SHI; EBERHART, 1998a)

In other words, the velocity vector of a particle will change on the influence of the best particular position it has achieved so far (p_{best}) and the best current position of the whole group (g_{best}). The new vector will be the resultant of all these vectors, changing the position of the particle. That happens to all particles of the population at each iteration. Every time a particle finds a new better spot, its p_{best} is updated, it also happens when one of them finds a new better global spot (g_{best}).

To better understand how the particle swarm algorithm works, it can be compared to a fish schooling like sociobiologist E. 0. Wilson referred. The whole fish school is looking for food and their position is defined by x, y and z in a three dimensional space underwater. They can sense how close they are to food (fitness) and they communicate with each other. When one of the fish comes closer to food, it sends a message to the school like: “I got closer, my position is the new g_{best}.” The others than change the direction they were heading to a new one, turning closer towards the new global best position. But this turning also depends on the best position closer to food that the fish remembers it has passed, so it also has its own local search.

To define the best solution at each iteration, the fitness must be calculated. To this problem of the optimization of the GAPID, the fitness evaluation process is the same as the genetic algorithm, explained in section 2.7.2.2 and uses the same fitness equation (17). In case of the fish schooling, is easy to picture the school converging to an optimal location in a three dimensional space (updating x, y and z), but in the case of the GAPID the parameters are not x, y and z, but the 8 or 6 gain parameters explained in section 2.4. So it is like the algorithm is moving in a 8 or 6 dimensional space, looking for a better combination of controller parameters.

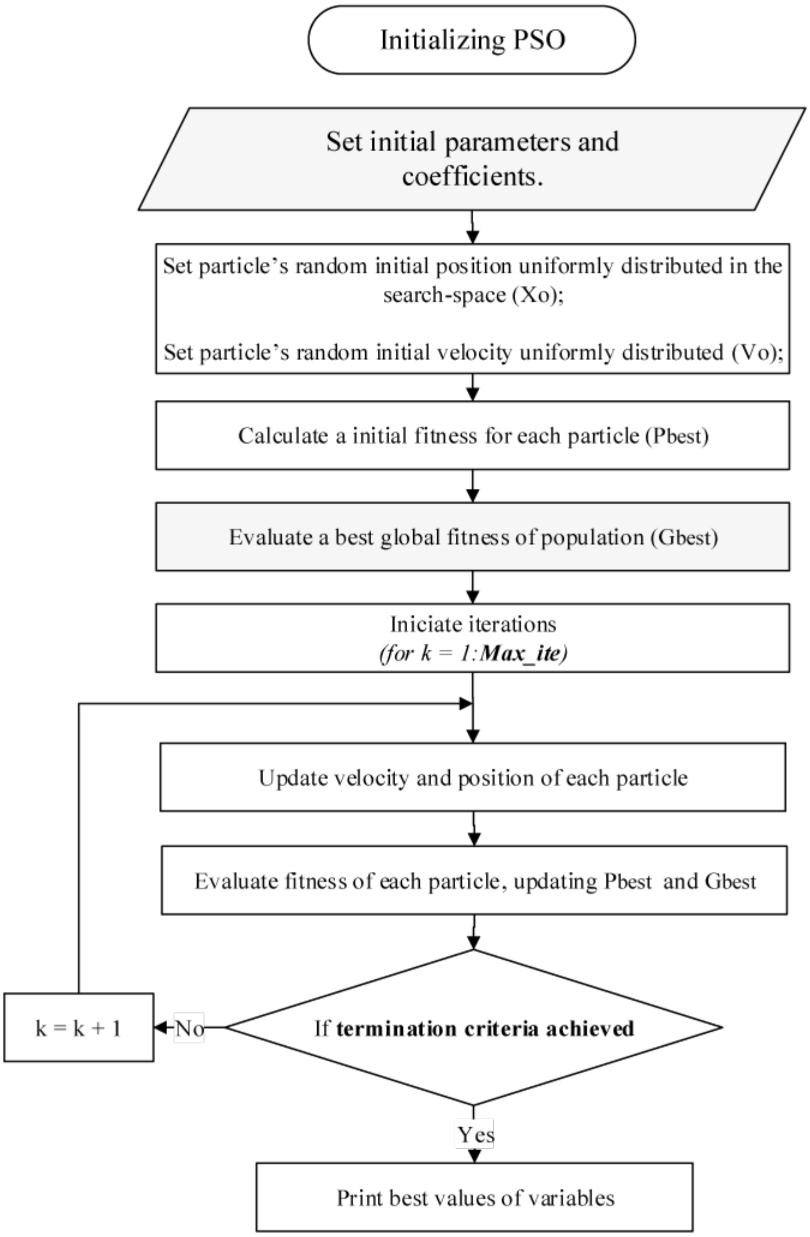
The general steps of the original PSO are summarized in a flowchart in Fig. (12). The termination criteria can be the maximum number of iterations or the given precision parameter of the best swarm fitness, whichever comes first (CLERC; KENNEDY, 2002).

One main difference between the GA and the PSO is that the PSO is more dynamic, while the GA is more static. On the first, every new iteration or generation, is a new group of individuals that carry the genes from their parents a generation before. Therefore, the children can represent a location further away from their parents, in one iteration the particles can “jump” to a new location. On the PSO, there is no breeding, the particles have velocity, taking steps and changing their own position every iteration.

In the PSO algorithm the definition of a topology is mandatory. This information determines how the particles exchange their position information (and fitness) at each iteration. In this investigation we address two proposals, global and ring topologies. It can strongly influence the search capability (KENNEDY; MENDES, 2002).

In the global topology, the position of each particle is shared with the entire swarm. In this work, the best fitness (g best) is related to the particle that present a set of parameters which leads to the best performance of the controller (PUCHTA et al., 2016a). On the other hand, the ring topology uses a communication between the neighbors of the current particle, in order to set a local best position. It compares the fitness between groups of three successive particles, the current one and the immediate predecessor and successor (CASTRO, 2006). Figure (13) presents the two main PSO

Figure 12 – PSO Flowchart

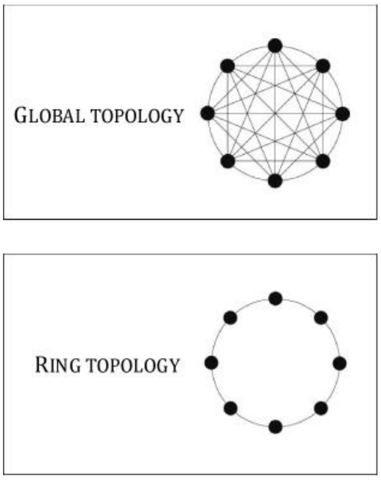

Initializing PSO
Set initial parameters and coefficients.
Set particle's random initial position uniformly distributed in the search-space (Xo);
Set particle's random initial velocity uniformly distributed (Vo);
Calculate a initial fitness for each particle (Pbest)
Evaluate a best global fitness of population (Gbest)
Iniciate iterations
(for k = 1:Max_ite)
Update velocity and position of each particle
Evaluate fitness of each particle, updating Pbest and Gbest
If termination criteria achieved
No
k = k + 1
Yes
Print best values of variables

source: own autorship

topologies.

It is possible to see from Figure (13) that while in the ring topology, the particles only communicate to the adjacent ones, on the global topology there is communication between all of them. The best position g best is saved in an auxiliary variable. It is important to remark that the algorithm structure utilized for both topologies can be the same.

Figure 13 – PSO topologies



source: own autorship

3 METHODOLOGY

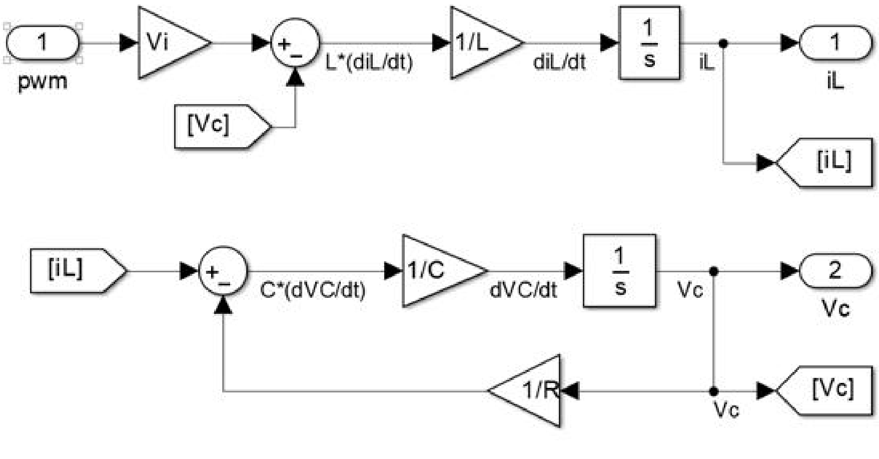
3.1 INTRODUCTION

This chapter will present the development of this work, how the simulation models for both the study and experimental plants were set up and how the optimization algorithms were applied to tune the GAPID for both cases. The process of building the experimental prototype plant will also be stated, along the PLC programming for the PID and GAPID. How and which results were captured for further analyses and how those analyses and results were measured and compared are included as well. The results and comparisons are presented in the next chapter Results.

3.2 DEVELOPMENT OF STUDY SIMULATION PLANT (DC-DC CONVERTER)

At first, a simulation model for the DC-DC Buck converter was developed based on its mathematical model explained in section 2.2. Fig. (14) presents this simulation model where the input is a pulse width modulation signal which controls how much power is delivered to the system from the constant input Voltage Vi.

Figure 14 – DC-DC converter simulation model



source: own autorship

In Table (1) are all the chosen parameter values of the converter.

A model of the GAPID was developed, alongside with a PID. They are responsible to set a duty cycle to the pwm, as a result, controlling the output voltage. Fig. (15) shows how the simulation

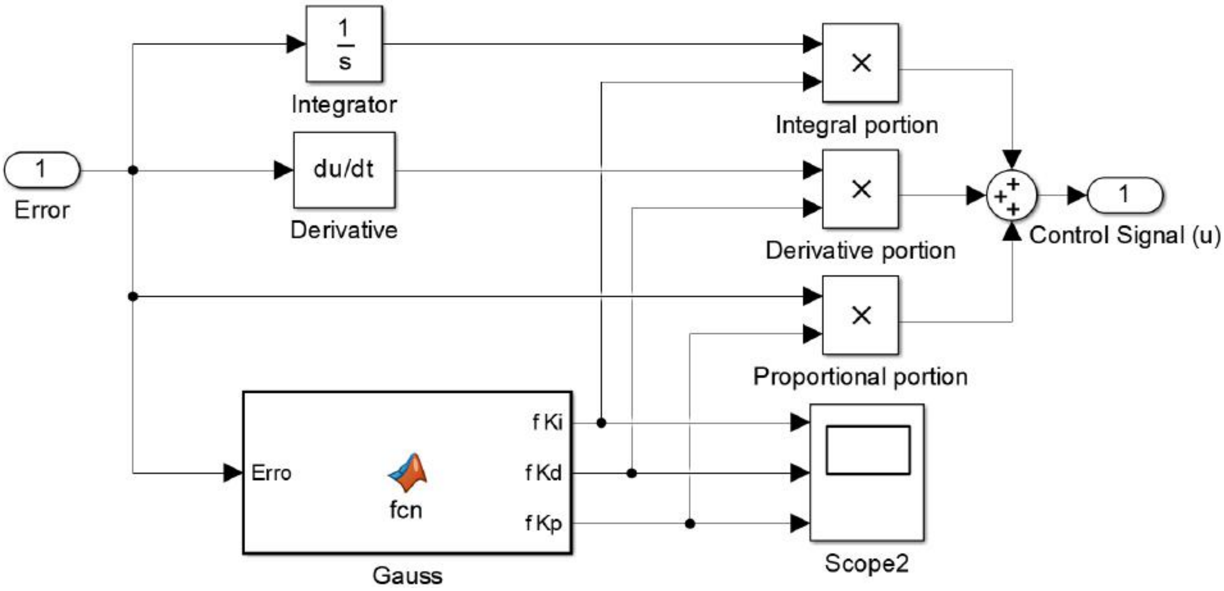
Table 1 – Buck parameters

Parameter	Value
Input voltage (V_i)	48 V
Capacitance (C)	10 μF
Inductance (L)	3 mH
Load resistance (R)	55 Ω
PWM frequency (f_{pwm})	50 kHz
Set-point voltage (target value)	30 V

source: own autorship

model of the GAPID was built.

Figure 15 – GAPID block



source: own autorship

The Gauss block contains the three Gauss functions for each branch of the PID (proportional, integral and derivative) represented by f_{Kp}, f_{Ki} and f_{Kd} respectively as follows in Eq.s (22),(23) and (24).

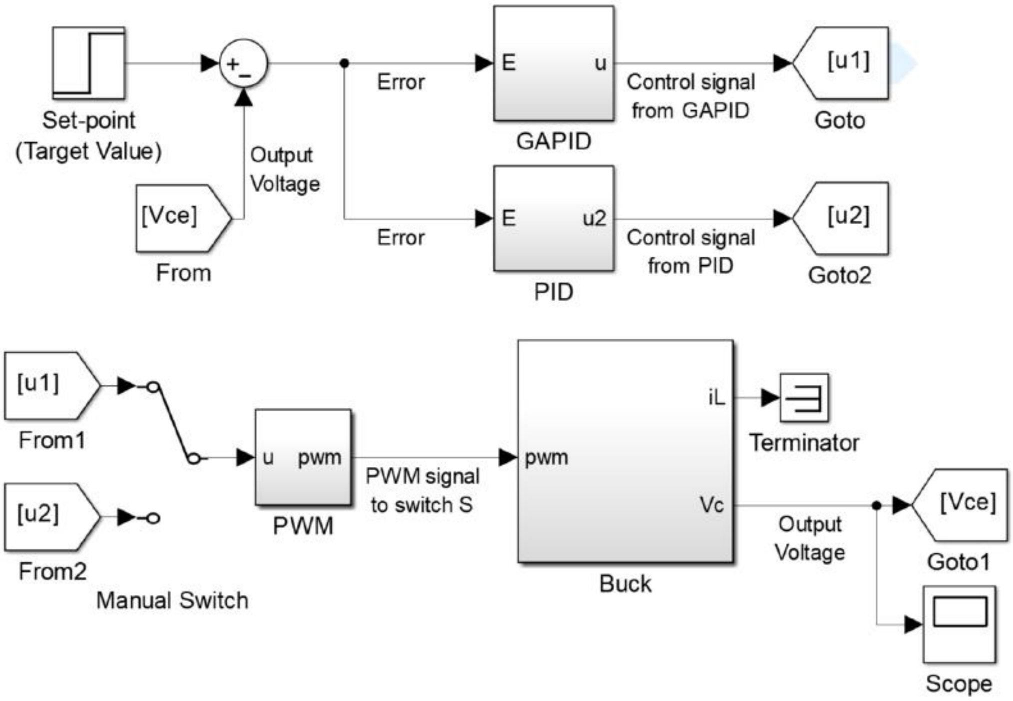
$$f_{Kp}(\sigma) = k_{p1} - (k_{p1} - k_{p0})e^{-q_p.\sigma^2} \tag{22}$$

$$f_{Ki}(\sigma) = k_{i1} - (k_{i1} - k_{i0})e^{-q_i.\sigma^2} \tag{23}$$

$$f_{Kd}(\sigma) = k_{d1}(1 - e^{-q_d.\sigma^2}) \tag{24}$$

The result of each function is multiplied by the error, the error derivative and its integral signal, all summed up to finally generate the control output signal u. The GAPID or the PID are linked to a pwm signal generator block and the Buck converter model to result in the overall closed-loop simulation model shown in the following Fig. (16).

Figure 16 – Complete simulation block of the Buck converter



source: own autorship

The GAPID and the PID do not run simultaneously. Which control method is currently being used can be chosen by a manual switch. The controlled output voltage (Vce) is subtracted from the set point to generate the error signal for the controller, closing the loop. A scope captures the output Voltage signal. Both Genetic Algorithm and PSO are in charge of finding the parameters values of the three Gauss functions. To do so, every iteration of the algorithms run the simulation model for 4ms. The output Voltage signal from the scope for this 4ms duration is used to calculate the fitness of these set of gauss function parameters rating how good they are. Six variants of the Genetic algorithm were tested using the test simulation model. These six variants were:

- GA 1: Original GA using roulette wheel with 70% of crossover rate.
- GA 2: Same as GA 1 but the crossover rate is 100% (always happens).
- GA 3: The selection used was the no death binary tournament.
- GA 4: Here, the “death tournament” was performed as selection.
- GA 5: A variation of GA 2, the roulette is applied after the mutation.

- GA 6: Same as GA 5, but the tournament is used instead of the roulette.

All variants had a population of 50 individuals and a maximum of 50 iterations. The best one is chosen by the average fitness given from a boxplot graphic. Another six variants were tested, but now of the PSO. They were divided in two main groups, one using the global topology and the other using the ring topology. They were developed as follows:

- PSO 1: This is the original proposal from Kennedy and Eberhart (KENNEDY, 1995), the one which uses Eq. (20) to calculate the velocity. The global topology is addressed.

- PSO 2: Another proposal which uses the global topology. In this PSO variant, a fixed inertial weight (ω) multiplies the previous velocity v_i in Eq. (20), generating Eq. (25) (EBERHART; SHI, 2000). The inertial value is a real value in the range between 0 and 1 (VESTERSTROM; THOMSEN, 2004). For a global fitness searching, large values of inertia are effective. Small values are should be utilized for local search (KHAN et al., 2012). This variable ensure a controllable particle's velocity, limiting its movement.

$$v_i(t+1) = \omega v_i(t) + r_1 c_1 (p_{best_i}(t)–x_i(t)) + r_2 c_2 (g_{best}(t)–x_i(t)) \quad (25)$$

- PSO 3: In this case, the inertial value decreases linearly at each iteration, from a maximum (w max) to a minimum value (w min), according to Eq. (26). This procedure provides a more efficient way for converging to a solution by limiting the particle's movement at each iteration (SHI; EBERHART, 1998b). Again, the global topology is addressed.

$$\omega = \omega_{max} - k\frac{(\omega_{max} - \omega_{min})}{ite_{max}} \quad (26)$$

  where ω_{max} = Maximum inertial weight value; ω_{min} = Minimum inertial weight value; ite_{max} = Maximum number of iterations; k = Index of iteration.

- PSO 4: This variant uses the same idea of the PSO 1, but utilizes the ring topology. Therefore, it does not have inertial weight to limit the velocity (CASTRO, 2006).

- PSO 5: The fixed inertial weight strategy is applied on this methodology, but using the ring topology. It is similar to the PSO 2 (EBERHART; SHI, 2000).

- PSO 6: Now addressing the ring topology, once again the linear inertial weight decreasing strategy is used, as in PSO 3 (SHI; EBERHART, 1998b).

Just like the GA, the PSO variants are evaluated by their average fitness from its boxplot graph.

The results of tests with both optimization algorithms are compared to a linear PID. The best variant of the better of the two algorithms is selected for the load sweep variation test.

In the load sweep, instead of running for a single load, the algorithm calculates the fitness of a specific set of parameters for 10 different loads ranging from 10 to 100 Ohms in steps of 10. The fitness for each load is summed up and divided by 10 (mean) as shown in Eq. (27).

$$Fitness = \frac{1}{10}\sum_{i=1}^{10}\frac{Fit_1 + Fit_2 + Fit_3 + ... + Fit_{10}}{10} \tag{27}$$

It is calculated for each individual of the algorithm, which represents a set of gauss parameters. As a result, the algorithms reaches a possible optimal robust solution. After this solution is found, it is tested and compared to the linear PID. For each load an output voltage result is registered. As the PID controller is not developed with any sort of adaptation method, it is expected the GAPID to achieve better results.

The controller must send a duty cycle as signal and for that it is between 0 and 1. Values beyond this range is not interpreted by the pwm and it ends up sending 1 or 0 anyway. This effect is control saturation. Next step was to apply a saturation sweep test.

In order to test and analyze this control saturation for the GAPID, it was put through a gain sweep. The optimization algorithm found the GAPID parameters for each gain and the sweep was done increasing its value by 10 every step, for 20 steps. A graph for each step is registered.

To measure how much the signal saturates, a saturation index was calculated similar to the IAE. The area inside the range between 0 and 1 was stripped, leaving only the saturation area graph. The saturation index is the IAE of each of these stripped curves and then turned into a fitness as in Eq. (17).

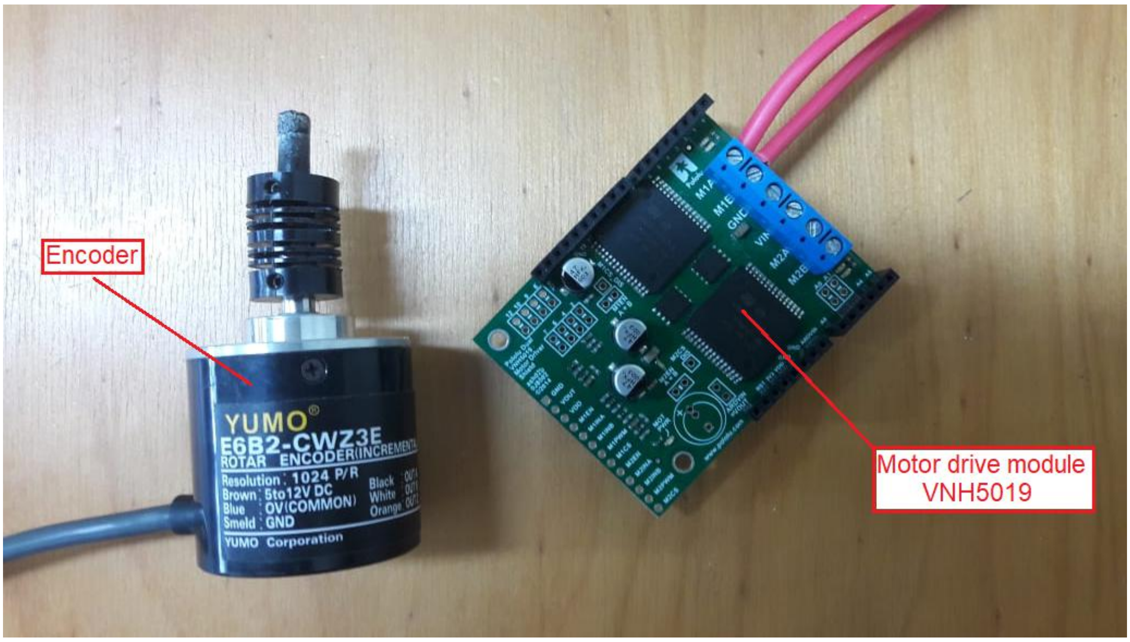
The linear PID is also put through the same gain sweep test and the fitness of each of the 20 stripped curves is calculated. With the 20 fitness from the GAPID algorithm and PID a graph with the ratio GAPID/PID Fitness is drawn

to demonstrate how the GAPID compares to PID as it comes close to saturation.

3.3 DEVELOPMENT OF THE EXPERIMENTAL PLANT (DC MOTOR)

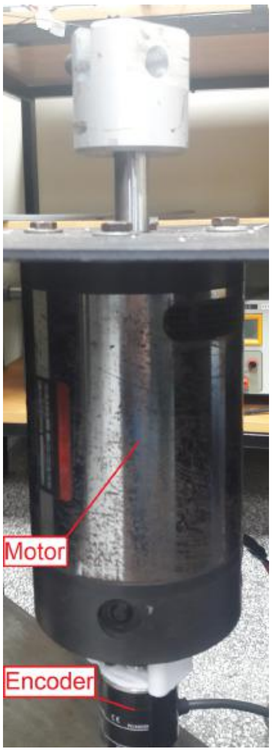
The experimental plant was built with a DC Motor powered by a controlled voltage source passing through the a Pololu Dual VNH5019 motor driver shield module. This module is a compact breakout board for ST's high-power motor driver IC, a fully integrated H-bridge to control the speed of a single brushed DC motor (POLOLU, 2019). The motor is connect to an encoder to measure its rotation speed. Fig. (1) presents the image of both the motor drive module and the encoder used and Fig. (2) is the picture of the Motor with the encoder attached.

Photograph 1 – Encoder and DC motor drive module



source: own autorship

Photograph 2 – DC motor



source: own autorship

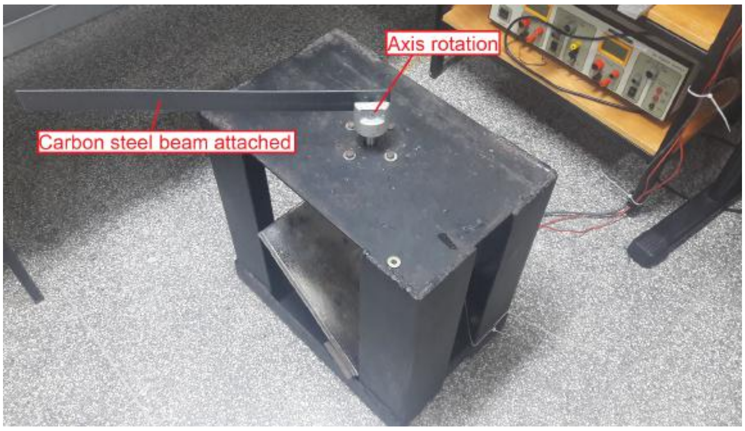
Data from the encoder is sent to a PLC from national instruments, the CompactRIO system with a multi function series C module. This system consists of a controller with a microprocessor and a FPGA programmable by the user. The module with analog and digital inputs and outputs connect the sensors to the PLC, in this case, the signal from the encoder as a digital input and the pwm signal to control the motor speed as an analog output (INSTRUMENTS, 2019). Fig. (3) illustrates the PLC used.

Photograph 3 – Programmable Logic Controller (PLC)

source: own autorship

Fig. (4) shows the system with the carbon steel beam attached to the rotation axis.

Photograph 4 – DC motor system

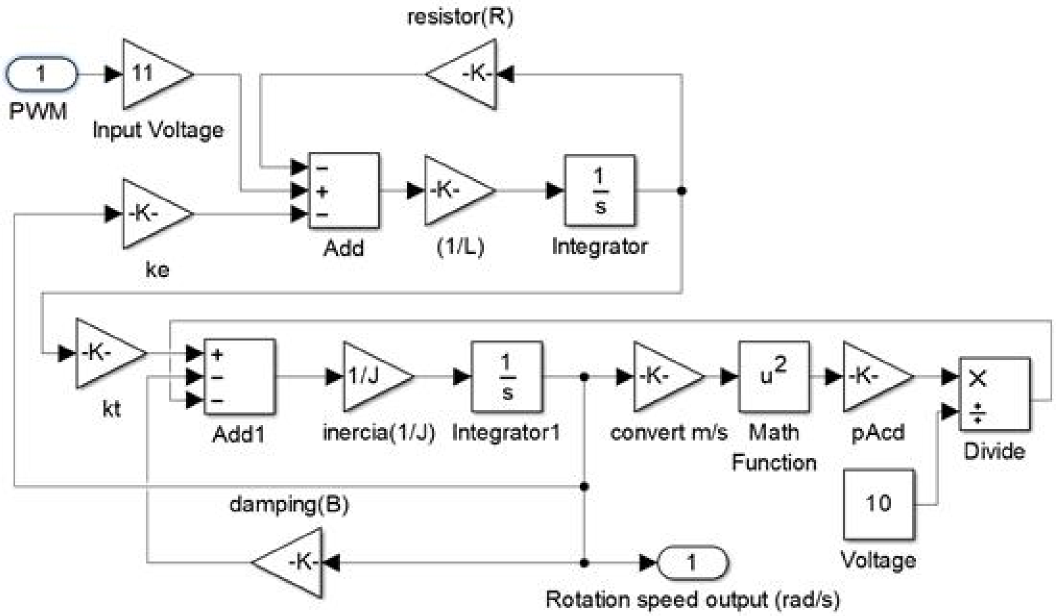


source: own autorship

Following the same steps from the study plant, a simulation model for the DC motor was developed based on its mathematical expression as present in section 2.3 by Eq. (9) and Eq. (10).

Fig. (17) presents this model.

Figure 17 – DC Motor simulation model



source: own autorship

The input is the PWM signal with an input voltage of 11 V. The output and controlled variable is the rotation speed in rad/s. Table 2 presents the values used for the simulation and correspond to the experimental model as well. The blocks after the Integrator1 block represents the drag force given by Eq. (11) in section 2.3.

Table 2 – DC Motor parameters

Parameter	Value	Unit
R	4.57	Ω
L	0.0032	H
K_e	0.23309	$\frac{N.m}{A}$
K_t	0.23309	$\frac{N.m}{A}$
B	$1.405(10^{-3})$	$\frac{N.m.s}{rad}$
V_i	11	V

source: own autorship

The value of inertia (J) changes depending on the beam attached to the system and function as a load. In this work, three different beams were used and all of them have the same surface area. These values of inertia for each beam is in table 3. Fig. (5) is a picture of these three beams used.

Table 3 – Inertia values of the beams

Beam	Inertia value (J)	Unit
Light slim aluminum	0.0035	
Carbon steel beam	0.0077	$kg.m^2$
Heavy thick aluminum beam	0.008	

source: own autorship

Photograph 5 – Metal beams

source: own autorship

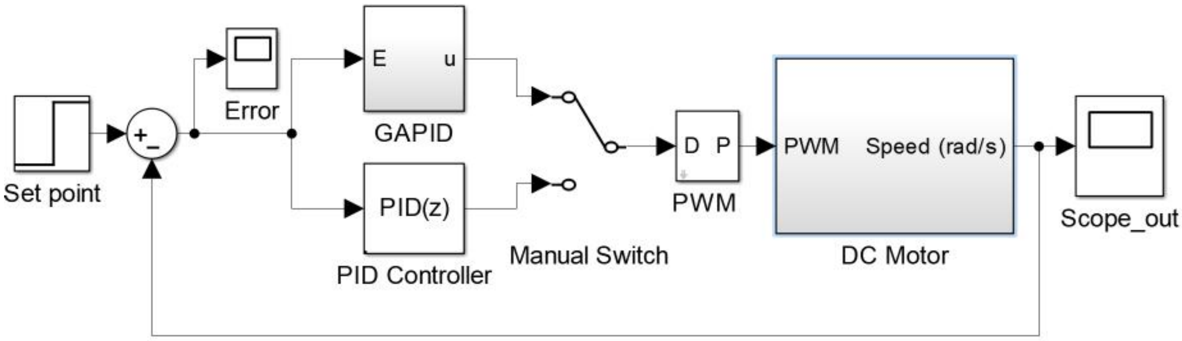
Although the input voltage could be set close to 90 V, for safety reasons the voltage input was kept at a maximum of 11 V due to the rotation speed it generates in open loop with the beams attached. The next step was to validate the simulation model. The system was tested without any load and with each of the 3 beams attached.

Without load and for the light slim aluminum beam, the input voltage was 11 V. For the other two beams the input voltage was 5 V to prevent any harm or damage caused by a fast rotation speed of the heavy beams.

After the validation of the model, a PID and a GAPID were develop and added to the simulation system. The GAPID was built the same way as for the study plant, the DC-DC Buck converter, as shown in Fig. (14) previously. As for the PID a simulink block was used. Fig. (18) shows the overall modeled system.

The output speed is subtracted from the set point, both given in rad per second, to produce an error signal used by both the GAPID and the PID. The control method is selected by a manual switch. The controller generates a control signal (from 0 to 1) and is transformed into a pulse width modulation signal to deliver the power to rotate the motor. The output signal is captured by the output scope.

Figure 18 – DC Motor simulation system



source: own autorship

Similar to the study plant, the optimization algorithm is in charge of finding the parameters values of the three Gauss functions for the GAPID by calculating the fitness from the output curve captured in the out scope every iteration.

Finished the simulations processes, the best parameters were applied to the PLC program for the experimental plant. Figures from 19 to 23 present the program develop inside the PLC to run both PID and GAPID.

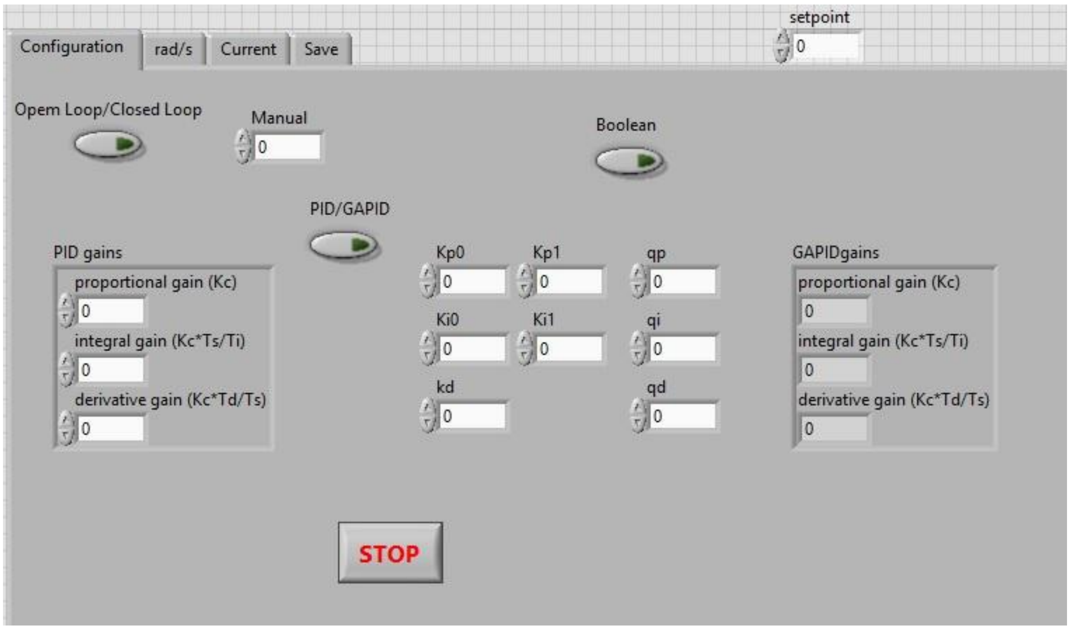
Fig. (19) is the interactive front panel of the program. This is where the values of gain for the PID and the GAPID are put. There is also a switch to select between open loop (manual) or closed loop (automatic) control. The automatic control can be switched from the PID and the GAPID. The manual value is a range from 0 to 1000 and represents the pwm output signal. A value of 600, for instance, indicates an on state of the pwm for 60% and an off time of 40%. It is basically a duty cycle of 0.6. The GAPIDgains block shows in real time the corresponding values of proportional, derivative and integral gain of the GAPID.

Fig. (20) is the background loop program for the encoder signal. The digital input signal from the encoder is captured and compared to the last signal. Because it is a digital signal, if it is higher than the last, it represents a rising edge, meaning the encoder just counted a pulse. In other words, it works as a rising edge detection. When a rise is detected, the condition box subtracts the current time monitored by the clock block from the last time a rising edge was detected (time taken between two pulses from the encoder). The encoder pulses 1024 times per rotation. As a result, the angle rotation of one pulse θ in rad can be calculated by Eq. (28).

$$\theta = \frac{2\pi}{1024} \cong 0.00613 \tag{28}$$

The clock time is given in milliseconds (ms), to change it to seconds (s) it is multiplied by

Figure 19 – Program front panel



setpoint 0

Manual 0

PID gains
proportional gain (Kc) 0
integral gain (Kc*Ts/Ti) 0
derivative gain (Kc*Td/Ts) 0

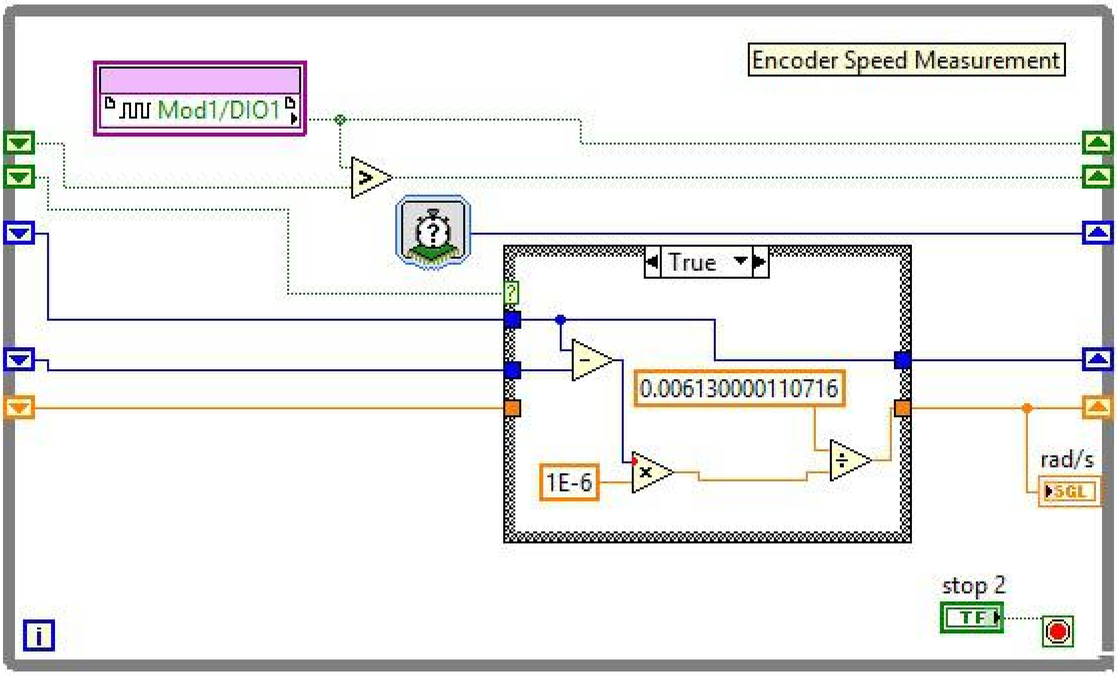
Kp0 0
Kp1 0
qp 0
Ki0 0
Ki1 0
qi 0
kd 0
qd 0

GAPIDgains
proportional gain (Kc) 0
integral gain (Kc*Ts/Ti) 0
derivative gain (Kc*Td/Ts) 0

source: own autorship

1.10^{-6}. The rotation of one pulse ($\theta \cong 0.00613$) is divided by the time between pulses in seconds. As a result, the output of the whole block is rotation speed (rad/s). Therefore this loop program turns the signal from the encoder into rotation speed.

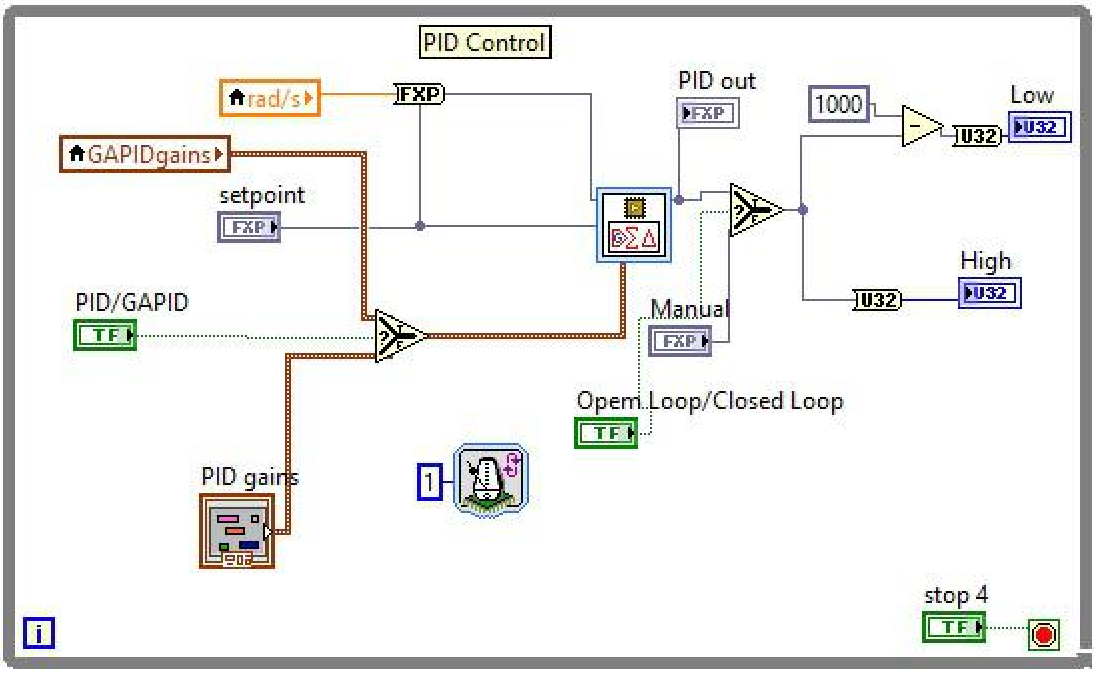
Figure 20 – Encoder programming



source: own autorship

Fig. (21) is the PID control loop program. The switch button from the front panel chooses between the PID and GAPID. The speed output information from the encoder loop is transformed to fixed point and goes to the PID block along side the set point. The switch button from manual to automatic is also seen here. When in automatic, the control output represents the high (on) pwm signal and it is scaled from 0 to 1000. When in 800, for example, the high value is 800 and the low value is $1000 - 800 = 200$. So it delivers 80% of power of the input voltage $11V$. It is important to note that this loop repeats every 1 ms, it is the controller clock. It is the same step time used for the simulation model.

Figure 21 – PID control program

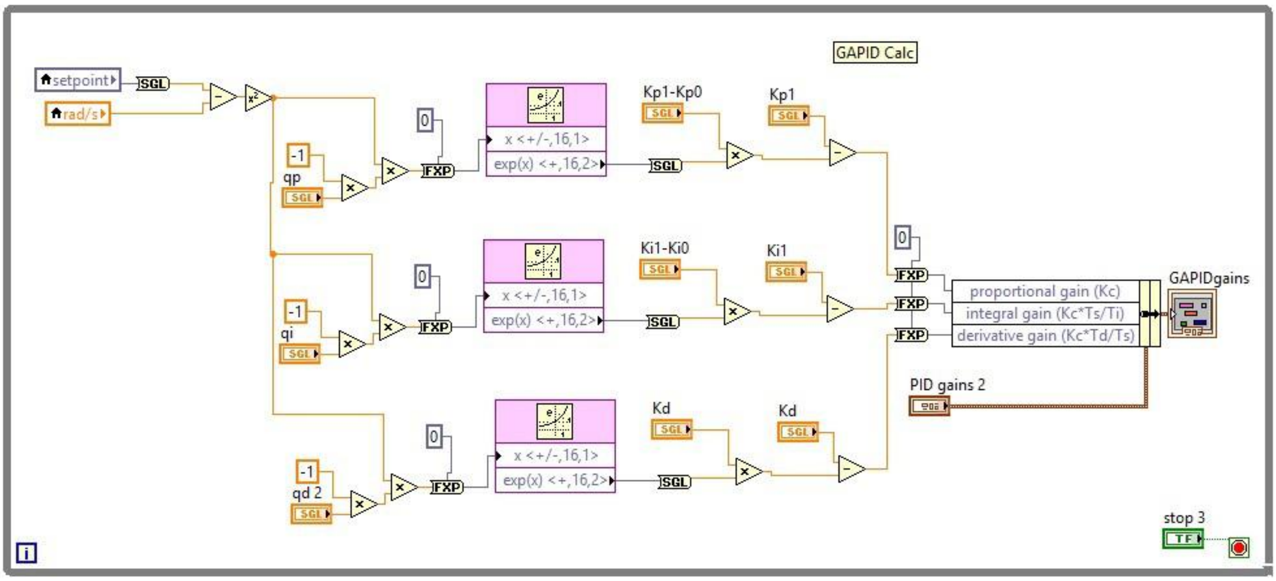


source: own autorship

Fig (22) is the GAPID control program. This loop is the gauss functions of the Eq.s (22), (23) and (24) depending on the error. The error (set point minus the measured value) is squared and multiplied by $-q$ then turned into a exponential to be multiplied by $k_1 - (k_1 - k_0)$. Then the value is transferred to a GAPID block.

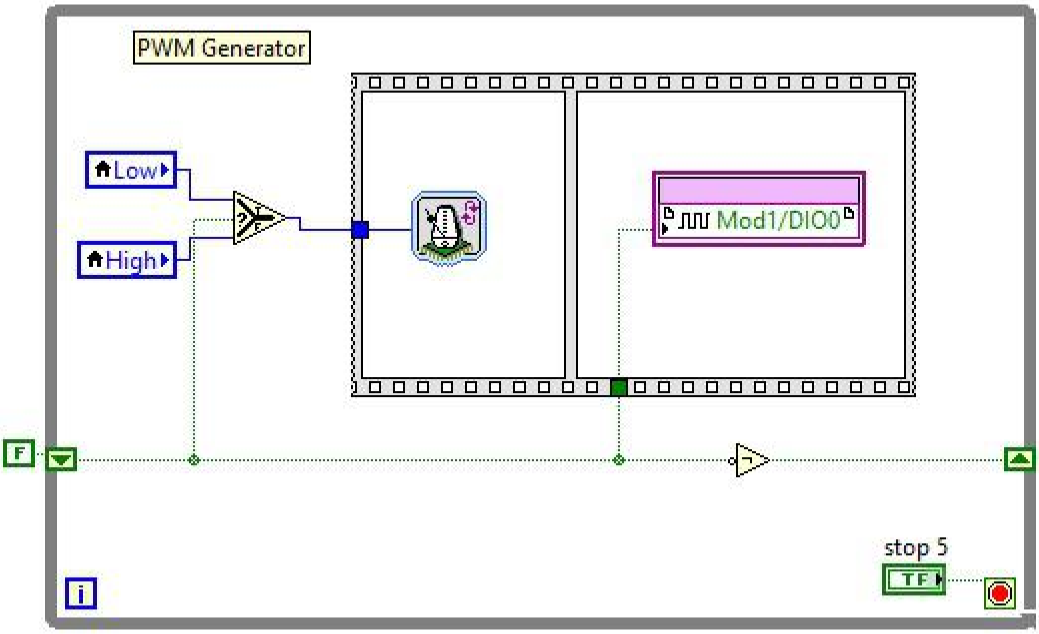
Finally, Fig. (23) is the pwm generator program. In this loop program a high (1) output signal is sent to the digital output of the PLC for the amount of time specified by the High variable coming from the control signal. After that it changes to low (0) output by the NOT logic port and this output is sent for the amount of time specified by the Low variable, generating a pwm signal to control the motor speed.

Figure 22 – GAPID control program



source: own autorship

Figure 23 – PWM generator program



source: own autorship

The controllers were tested for the three beams. A change of the set point from 0 to 5 was done to capture the behavior of the controllers in order to compare them.

4 RESULTS

4.1 INTRODUCTION

This chapter presents the results for both plants. First, is presented the results comparing six variants of the Genetic Algorithm (GA) and six variants of the Particle Swarm Optimization (PSO) used to optimize the GAPID with the link parameters method for the first plant, the DC-DC converter. Then, the best variant of each algorithm is compared in a addition to a linear PID. The best algorithm is used for the other tests, a load sweep test and a saturation test. At last, a comparison of the computer model and the real experimental model of the DC motor is shown to validate the computer model developed with each of the three beams attached. This model is used to run the best algorithm of the two (GA and PSO) to find the parameters of the GAPID. The parameters are applied to the real prototype model and a comparison between the GAPID and a linear PID with all three beams is shown.

4.2 GA VS PSO

The six variants of Genetic algorithms were tested as explained in section 3.2. The variants are changes on the selection method, crossover rate and the order of selection, mutation and crossover. Tests were done through simulations of the GAPID and the buck converter on the Matlab® software using the first simple IAE fitness function in eq. (17) as presented in sub-section 2.7.2.2. The GA variants 3, 4 and 6 achieved the best results (best fitness) using the tournament selection. Table 4 shows the best single fitness achieved by each variant.

Table 4 – Best fitness of each GA variant

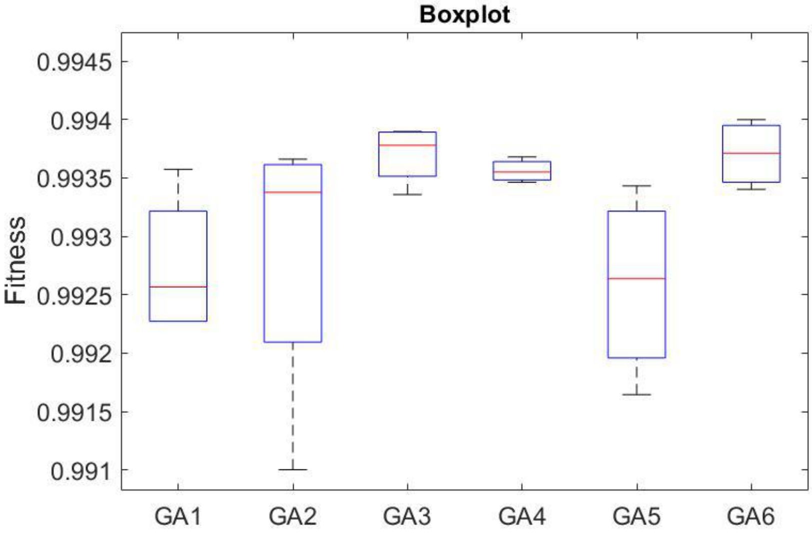
GA Model	Fitness	Ranking
GA 1	0.993574	5
GA 2	0.993661	4
GA 3	0.993898	2
GA 4	0.993680	3
GA 5	0.993432	6
GA 6	0.993999	1

source: own autorship

A box plot of the fitness of all six variants is shown in Fig.(24)

Analyzing the box plot, GA variants 3 and 6 achieved the best results. Although variant 6

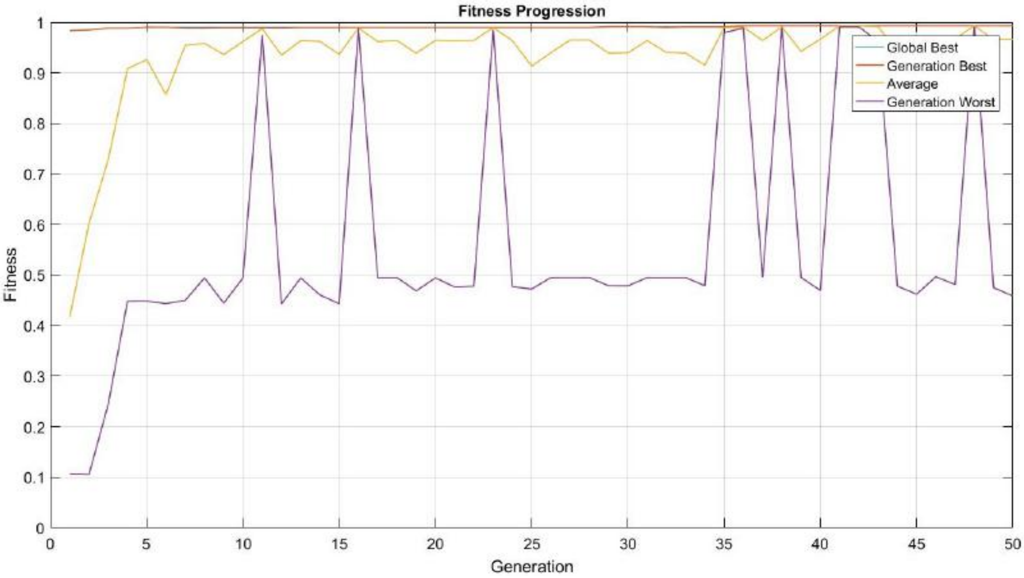
Figure 24 – Fitness box plot of GAs



source: own autorship

found the best single result, GA variant 3 average fitness is the best (represented by the red line in the box plot).

Graph 6 – GA 3 Fitness progression



source: own autorship

Figure 6 is GA 3 fitness progression and shows it converged quickly and maintained a high average fitness, probably due to the initial condition parameters.

For all those reasons, the GA 3 variant which uses the tournament selection and 100% crossover rate was used in this work and compared to the PSO.

A similar method comparing six variants of PSO was also done and tested as explained in section. The differences between the variants are the topology and addition of inertial coefficients. The testing method follows the exact same procedure of the previous test regarding the GA. PSO variants 1, 2 and 3 achieved the best results as presented in table 5.

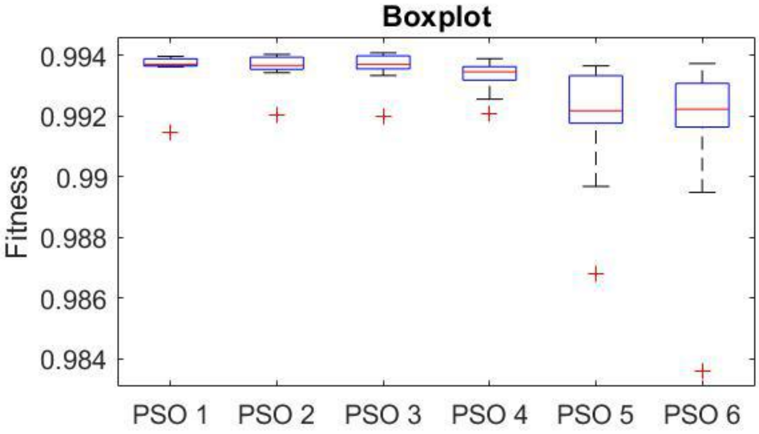
Table 5 – PSO variants average performance

PSO	Average fitness	Ranking
1	0.99353502	3
2	0.99355904	2
3	0.99358206	1
4	0.99327403	4
5	0.99176691	5
6	0.99141938	6

source: own autorship

The box plot in Fig.(25) of all variants provides further analysis to the results.

Figure 25 – Fitness box plot of PSOs



source: own autorship

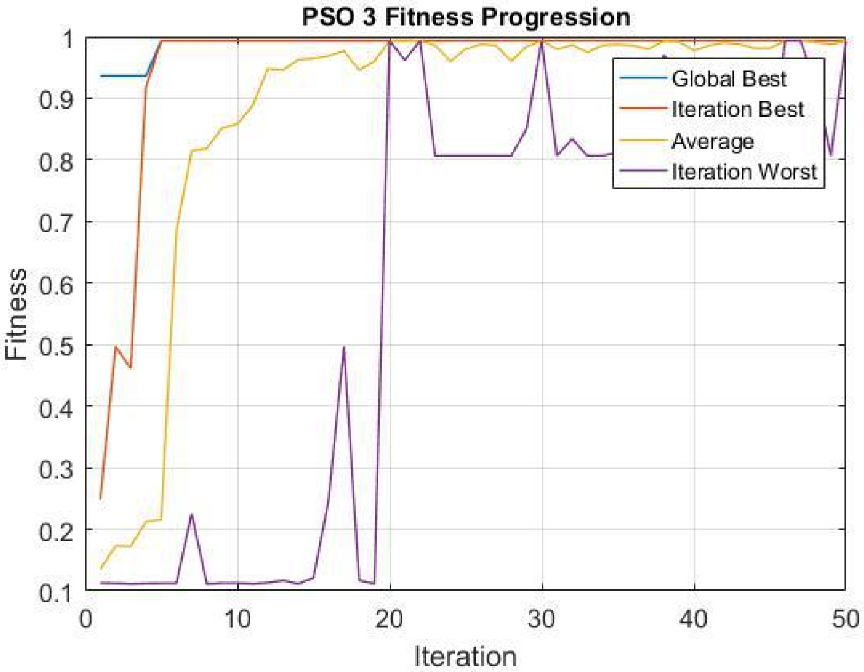
The three variants 1, 2 and 3 presented good results and all could be used as they would achieve similar results. PSO 3 was chosen in this study due to addition of a decreasing inertial coefficient ω to the velocity equation (20). Figure shows the PSO 3 fitness progression.

The addition of a decreasing inertial weight provides a more efficient way for converging to a solution by limiting the particle's movement at each iteration (SHI; EBERHART, 1998c).

Table 6 contains the values used for the coeficients in equation (25).

The inertial value is a real value ranging between 0 and 1 and it decreases linearly at each iteration, from a maximum (ω_{max}) to a minimum value (ω_{min}), according to eq.(26) explained in section

Graph 7 – PSO 3 Fitness progression



source: own autorship

Table 6 – PSO coefficient values

Coefficient	Value
$c_1$and c_2	2.05
Population	30
Maximum number of iterations	50
Search-space	100
ω_{max}	0.9
ω_{min}	0.4

source: own autorship

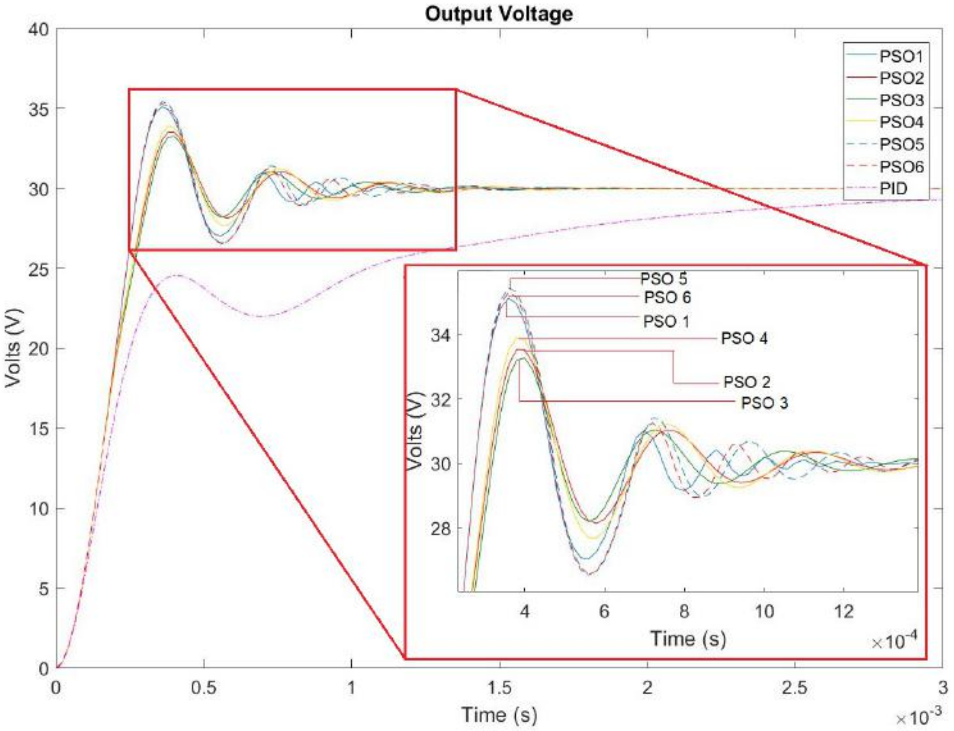
3.2

The simulation values of the parameters of the Buck converter as in Fig.(14) are in table 1. The output voltage of previously works, despite having good fitness values, presented high overshoot as shown in Fig.(8).

A reason would be because the IAE calculation is a combination of both overshoot and settling time. Consequently the algorithm possibly “overlooked” the overshoot and decreased the settling time to improve the fitness. It is due to the usage of equation 17. Taking it into consideration, the fitness calculation was changed to the new one as in equation 18. When the maximum voltage value surpasses the voltage target value 30 V it is saved and added to the IAE.

The previously chosen GA and PSO were compared on this simulation. The linked parameters method was used to relate the GAPID gains to the PID gains $K_p = 0.004455$, $K_i = 45$, and

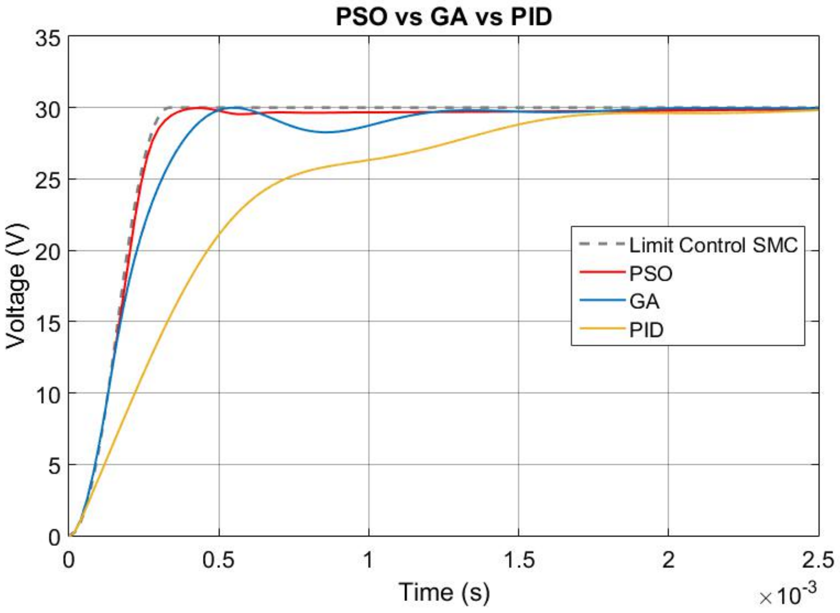
Graph 8 – GA and PSO output voltage



source: own autorship

$K_d = 1.9845\,10^{-6}$. Fig.(9) is a comparison graph from the best controlled output voltage by both methods and the PID.

Graph 9 – GA vs PSO voltage output



source: own autorship

Figure 7 is a table with the fitness that comes from the IAE of the 4 curves data from Figure 9.

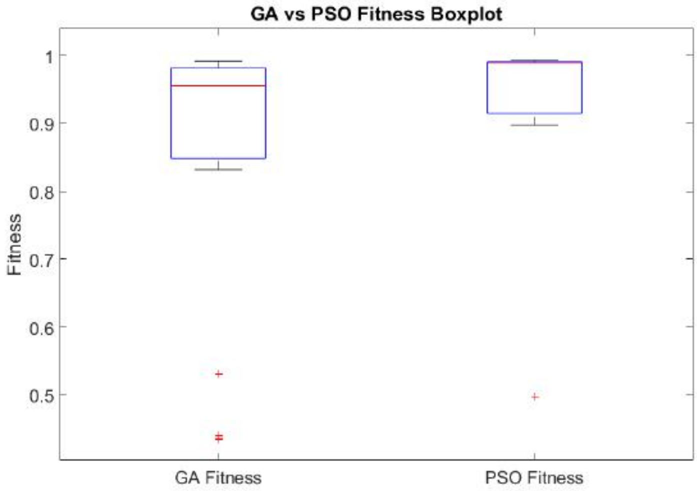
Table 7 – Best Fitness comparison

Control	Fitness	Ranking
Boundary Control	0.995232	1
GAPID with PSO	0.993866	2
GAPID with GA	0.992935	3
Traditional PID	0.986303	4

source: own autorship

Compared to the boundary control, the GAPID with PSO was the better of the 3 controllers achieving a best fitness of 0.993866 while GA's achieved 0.992935 and the PID fitness was 0.986303. Both optimization algorithms ran 30 times resulting in 30 different set of GAPID gains and fitness. Fig.(26) is a box plot of these 30 fitness.

Figure 26 – GA vs PSO fitness box plot



source: own autorship

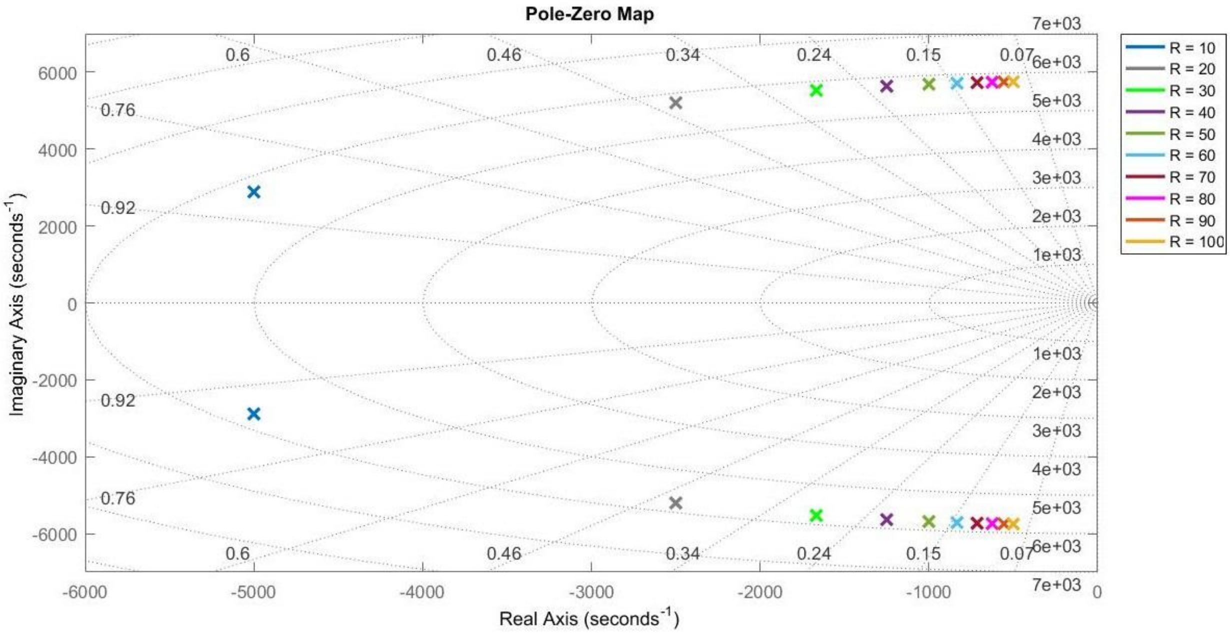
It is noticeable the better average fitness of the PSO over GA's.

4.3 LOAD SWEEP TEST

Next step was to do a load sweep to test the robustness of the GAPID compared to the PID. Fig.(10) shows a poles and zeros map of the Buck considering a load sweep from 10Ω to 100Ω.

The open loop poles of the Buck approaches the y axis which means the real part decreases and the imaginary part gets higher, increasing oscillation.

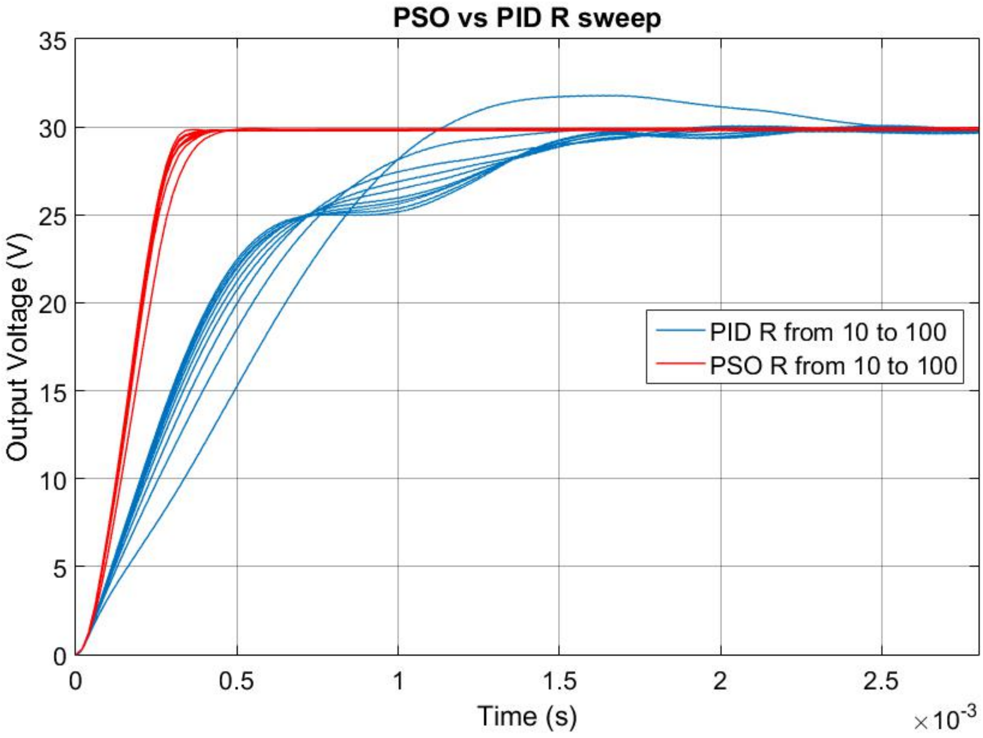
Graph 10 – Poles and zeros map for the Buck



source: own autorship

Fig.(11) is the output voltage graph comparison. The PSO algorithm was run 10 times adding 10Ω every step to find a single set of gains best suitable to all loads.

Graph 11 – Robust GAPID with PSO

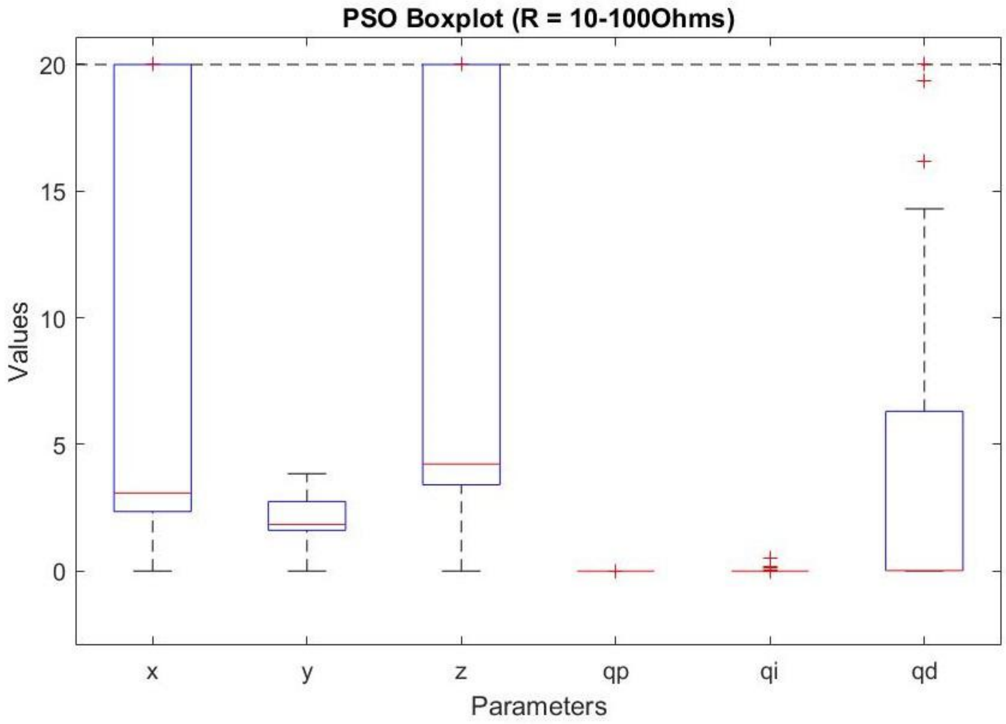


source: own autorship

Graph in Fig.(11) proves the adaptability over the PID in maintaing a better responses. As the linear PID was tuned to a load of 55Ω, it clearly starts to lower its control capability when the load increases or deacreases. The GAPID output, on the other hand, barely changed. The GAPID optimized by the PSO was 4.25 times faster than the PID in average.

Parameters for each specific load configuration was found as well and a box plot in Fig.(27) shows the parameters tendencies.

Figure 27 – Boxplot of each linked parameter



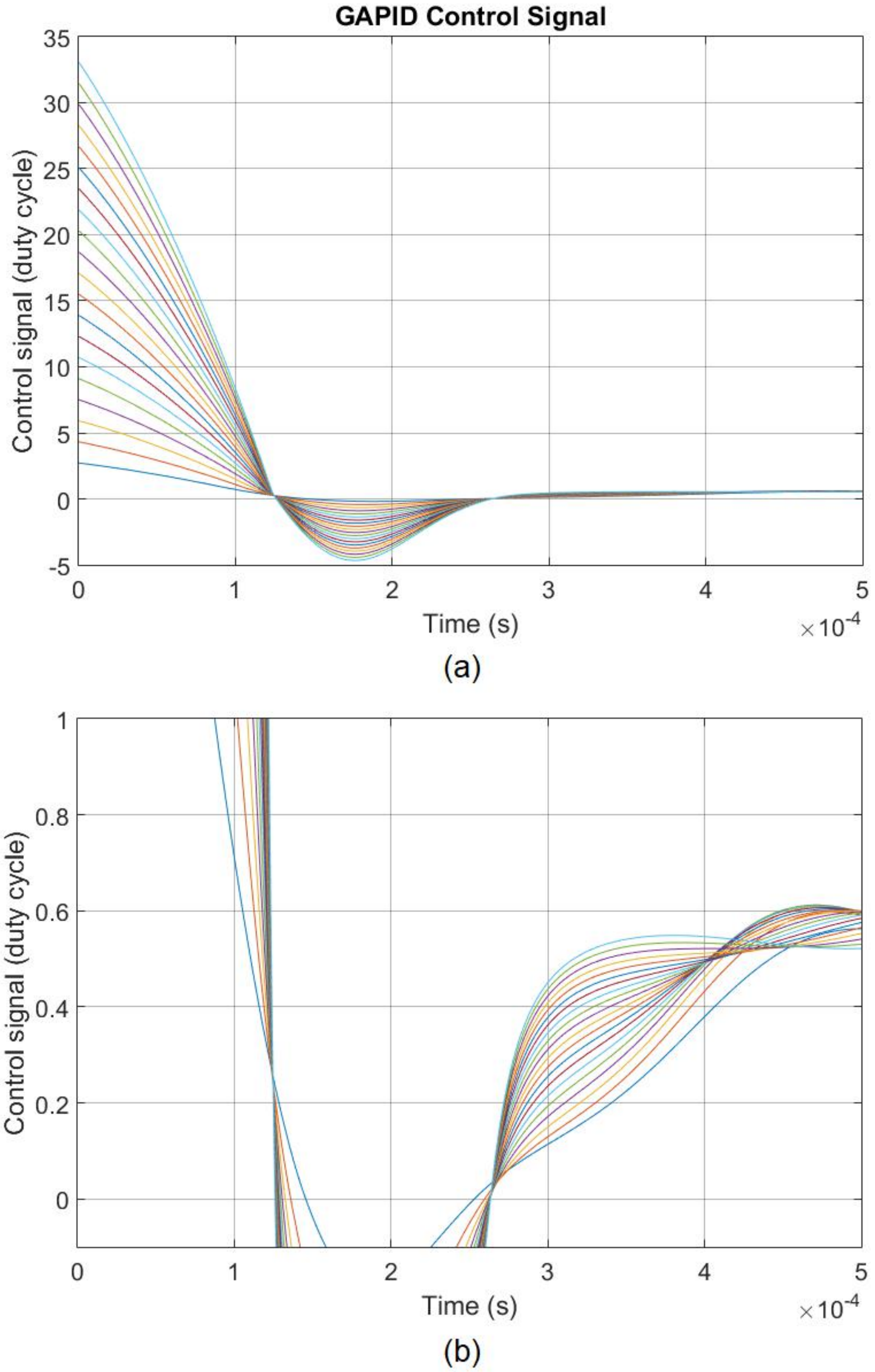
source: own autorship

In Fig.(27) it was expected to express a traceable tendency of the parameters in order to find a possible method to set and adjust the Gaussian variables without the aid of optimization algorithms. However it was not possible conclude due to the behavior of the parameters.

4.4 CONTROL SATURATION TEST BY GAIN SWEEP

In order to test and analyze the control saturation for the GAPID, it was put through a gain sweep to the PID it is linked. The PSO found the GAPID parameters for each gain and the sweep was done increasing it’s value by 10 every step, for 20 steps. Fig.(12a) shows the GAPID control signal as a whole without limiting the y axis, while Fig.(12b) limits the y axis from 0 to 1 to give an idea of the saturation.

Graph 12 – GAPID control signal



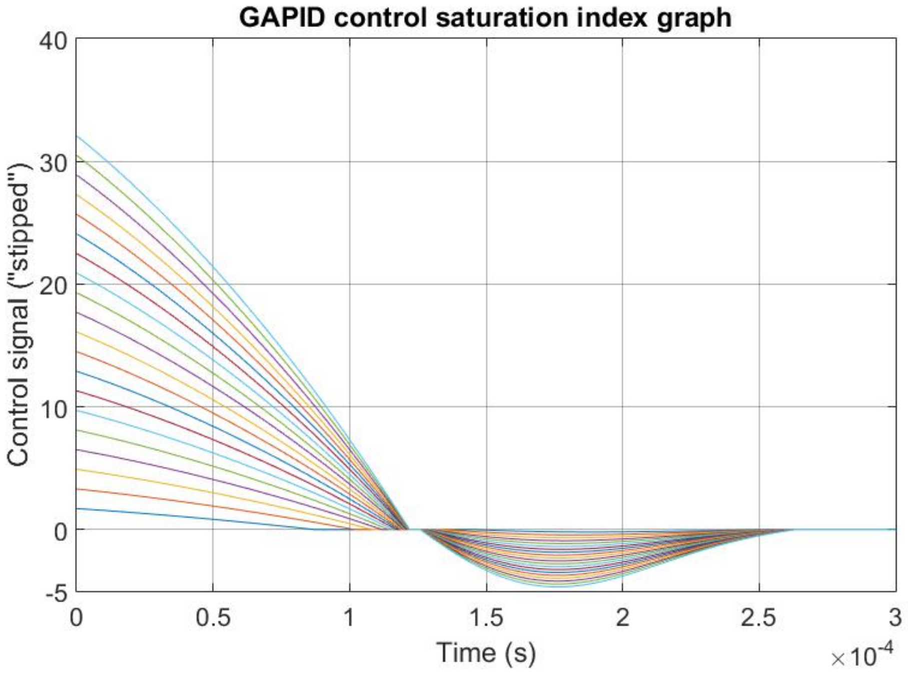
source: own autorship

The control saturates during the beginning but settles later on. Also, the saturation gets higher as the gain increases.

To measure how much the signal saturates, a saturation index was calculated similar to the

IAE. In Fig.(13) The area inside the range between 0 and 1 was stripped, leaving only the saturation area graph. The saturation index is the IAE of each curve.

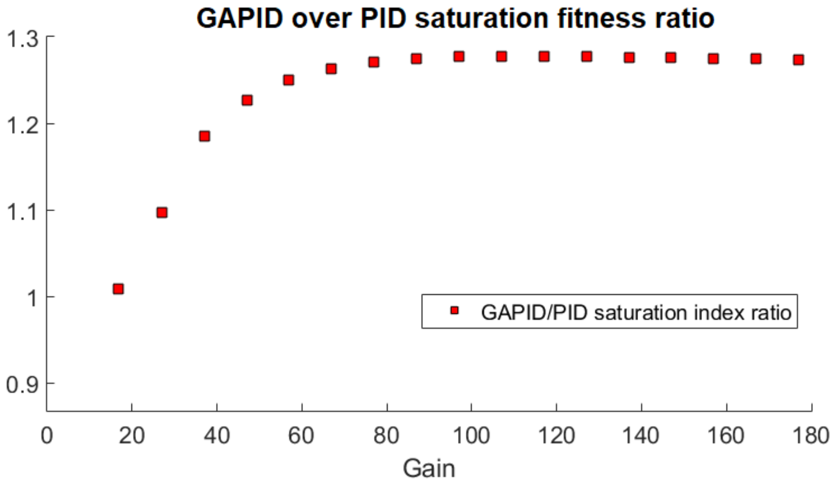
Graph 13 – GAPID control signal “stripped”



source: own autorship

A ratio of the fitness from GAPID over PID was done and ilustrated in Fig.(14)

Graph 14 – GAPID over PID saturation ratio



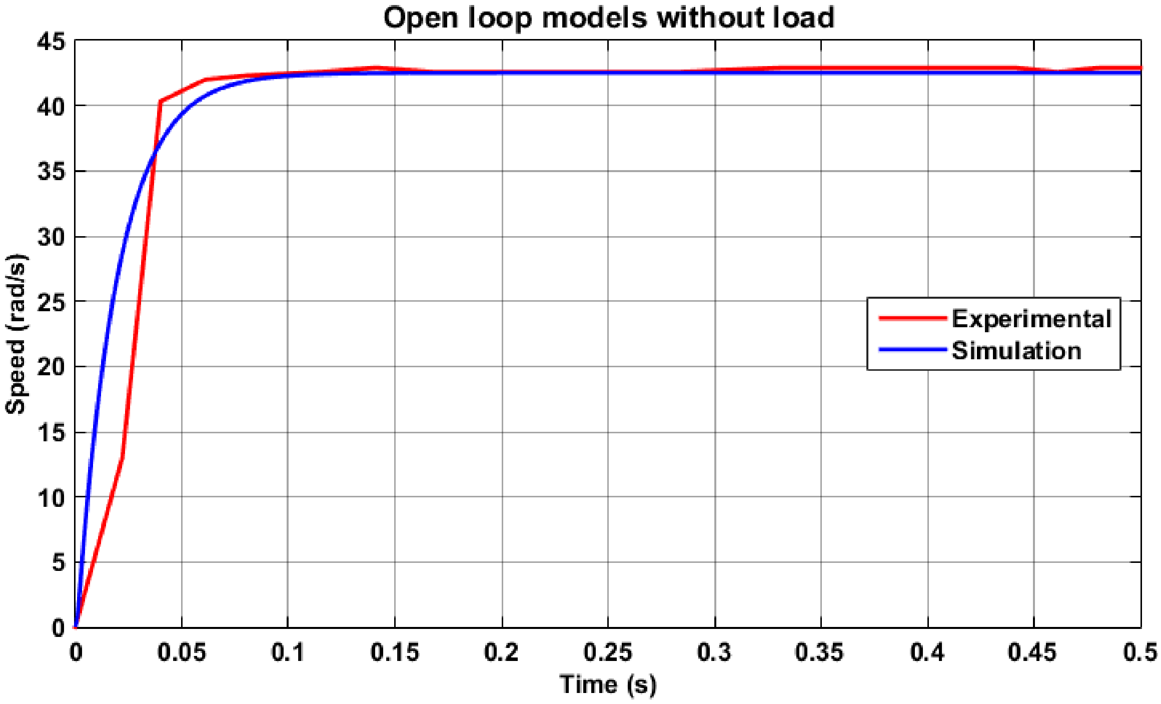
source: own autorship

When the PID gain is low the fitness of the PID is slightly worse than the GAPID, coming close to a 1 to 1 ratio. However, as the gain approaches the tuned PID value, the GAPID fitness gets better until it reaches a value where the GAPID presents no further improvement over the PID controller, reaching the maximum value of 1.28 ratio.

4.5 VALIDATION OF THE SIMULATION MODEL OF THE DC MOTOR PLANT

The validation of the simulation model for the DC motor was done by comparing the output graphs in open loop with the experimental prototype. Fig. (15) presents the graph without any beam attached.

Graph 15 – Open loop models without load

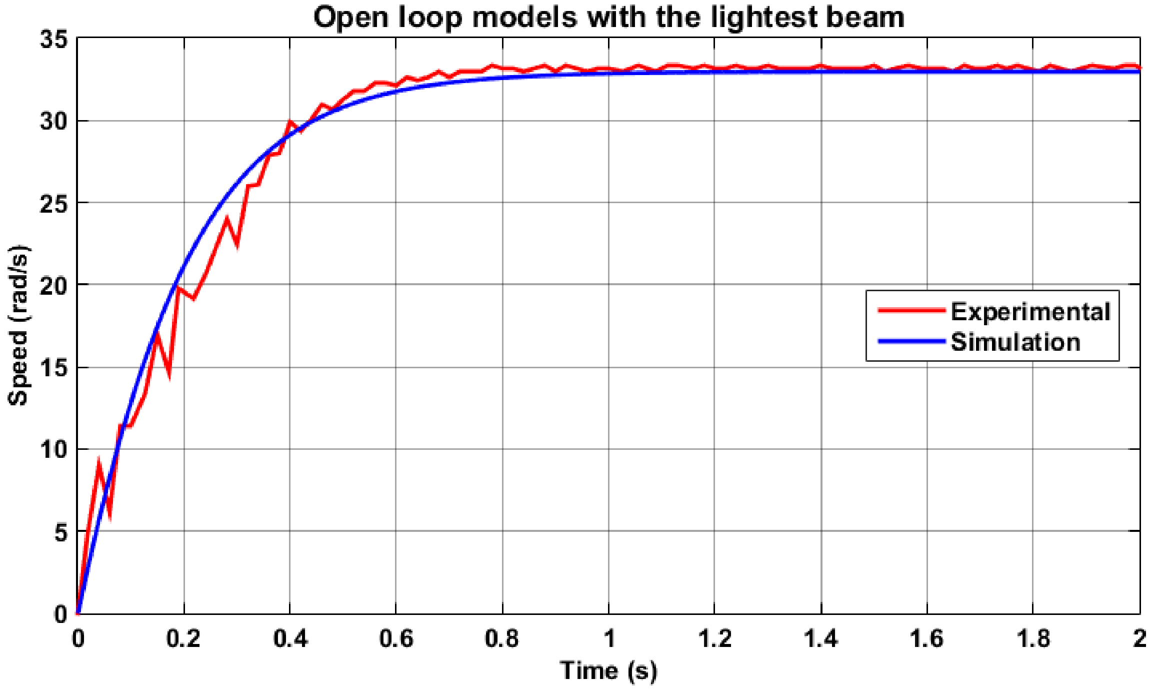


source: own autorship

The input voltage is 11 V. The rotation speed of both models show similar behavior and is enough to validate the simulation model without any load attached to the system. Next, Fig. (16) shows the graphs with the first light aluminum beam attached.

In Fig. (16) is also noticeable that both outputs graphs behave similarly. A difference from the output without any load is that the beam vibrates. Consequently, the experimental curve (in red) presents a behavior similar to a distortion or noise in the y axis (speed). One may also note that because of the addition of the beam as a load, even with the input voltage of 11 V, the maximum speed is approximately 33 rad/s, lower than the 42 rad/s of the system without any load.

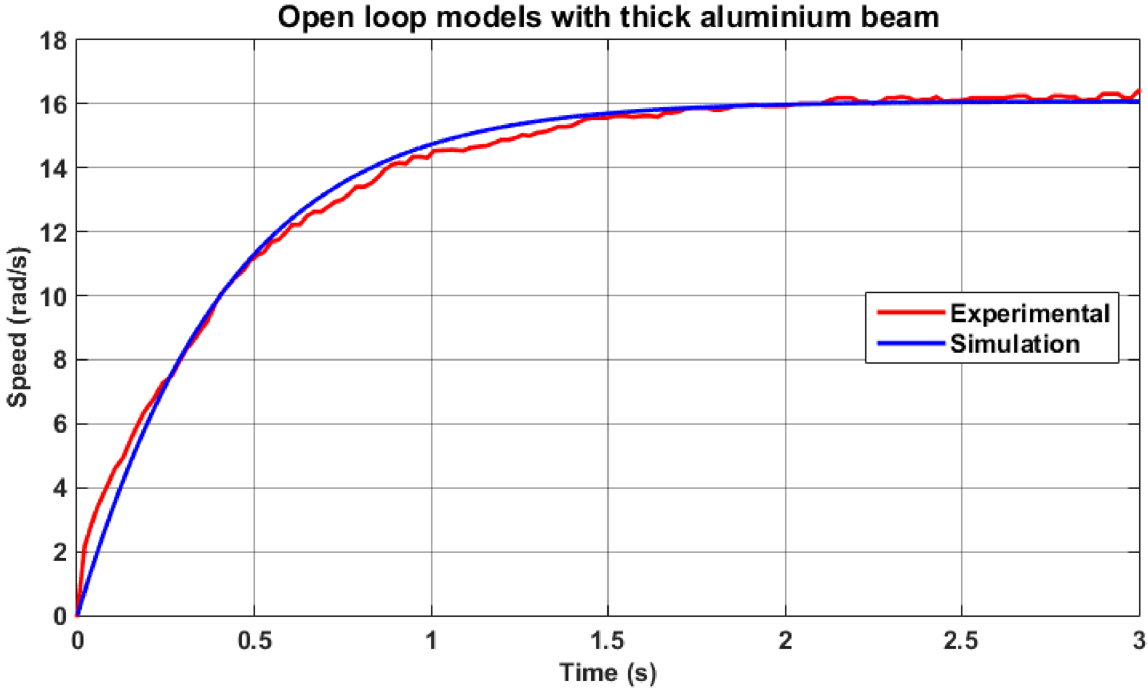
Graph 16 – Open loop models with the lightest beam



source: own autorship

The outputs from the following Fig. (17) were captured with the thick aluminum beam.

Graph 17 – Open loop models with thick aluminum beam

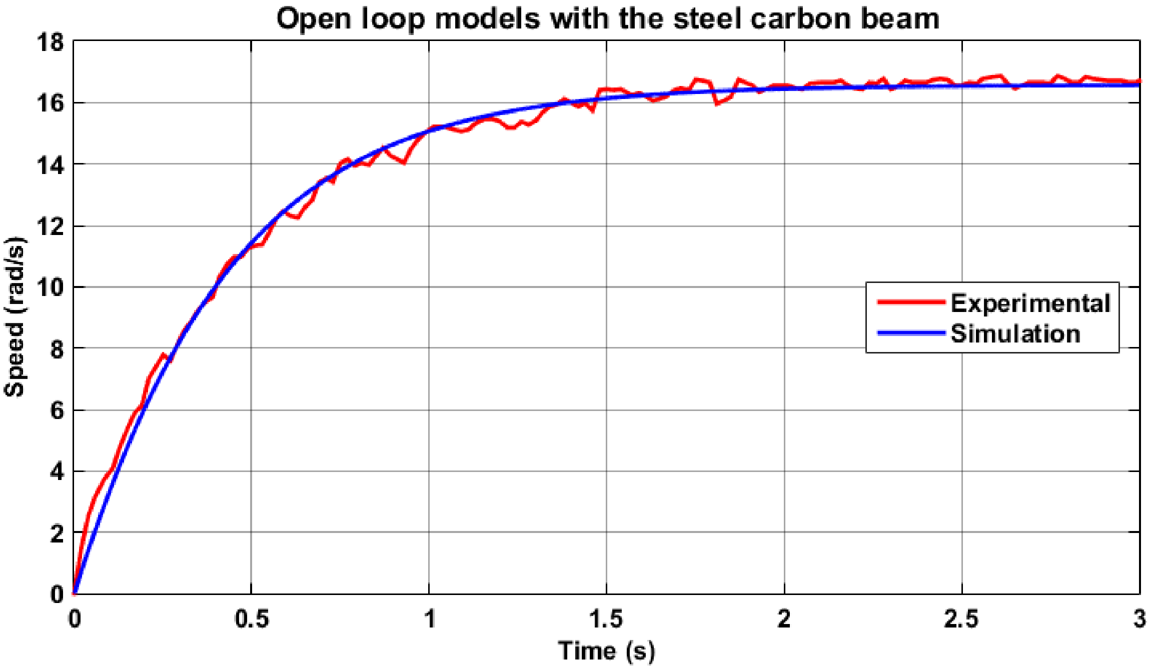


source: own autorship

In Fig. (17), once again, the behavior of the output of both models is the same. The input voltage to test the model with this thick aluminum beam was 5 V instead of 11 V. It was chosen so to safety reasons, preventing any harm or damage from the rotation speed with a heavier beam attached. Due to its heavier mass, and the slower rotation speed, the thick aluminum vibrates less.

Fig. (18) presents the outputs for the metal steel beam as load. A input voltage of 5V was applied for the same safety reasons. Both curves match enough again to validate the model.

Graph 18 – Open loop models with the steel carbon beam



source: own autorship

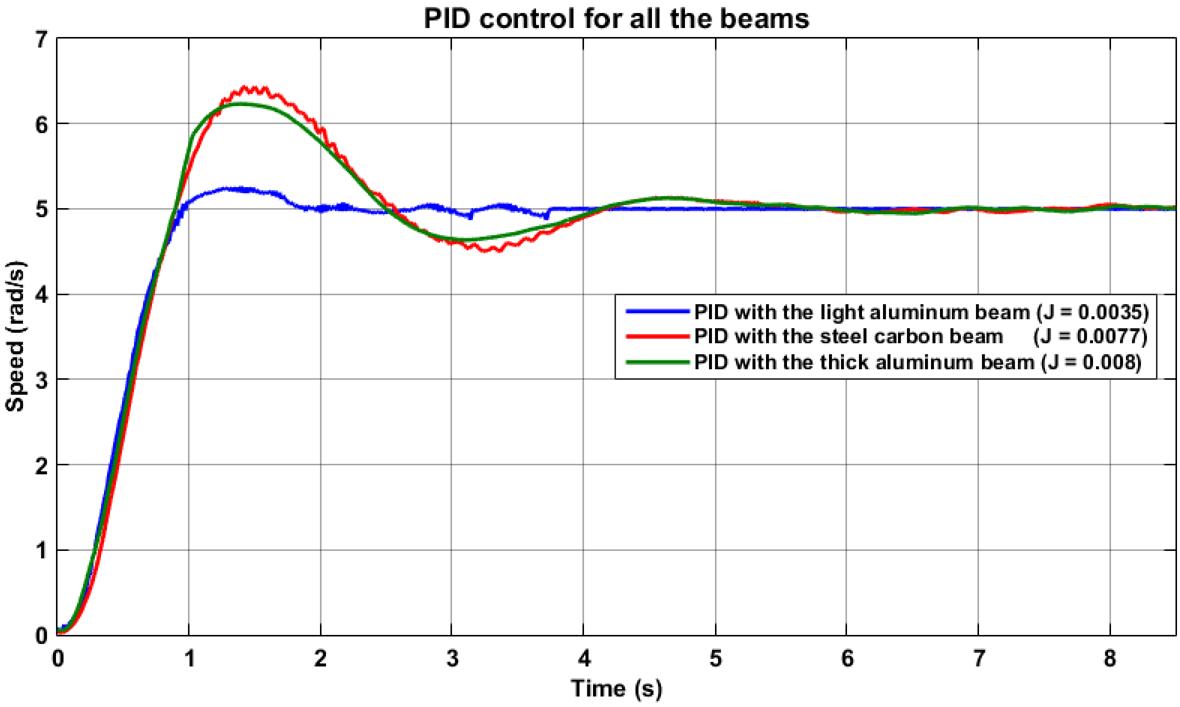
Comparing the models from their rotation speed output in open loop with each of the three metal bems attached, the simulation model corresponds to the experimental prototype model and is validated.

4.6 LINEAR PID VS GAPID COMPARISON FOR THE DC MOTOR PLANT

A PI controller was developed for the DC motor plant with the light aluminum beam attached. the absence of a derivative gain was chosen to prevent issues caused from noise. The values of gain were 0.2578 for the proportional gain k_p and 0.0625 for the integral gain k_i. the output response for each of the three beams is in Fig. (19).

The set point was 5 rad/s. In Fig. (19) the speed output kept an overshoot under 5% and a settling time under 2 seconds with the light aluminum beam it was projected for. In contrast, for the

Graph 19 – PID control for all the beams



source: own autorship

two other beams, it could not maintain a low overshoot and took longer to stabilize. The system was not unstable, the PID was still able to control, but not with the same performance.

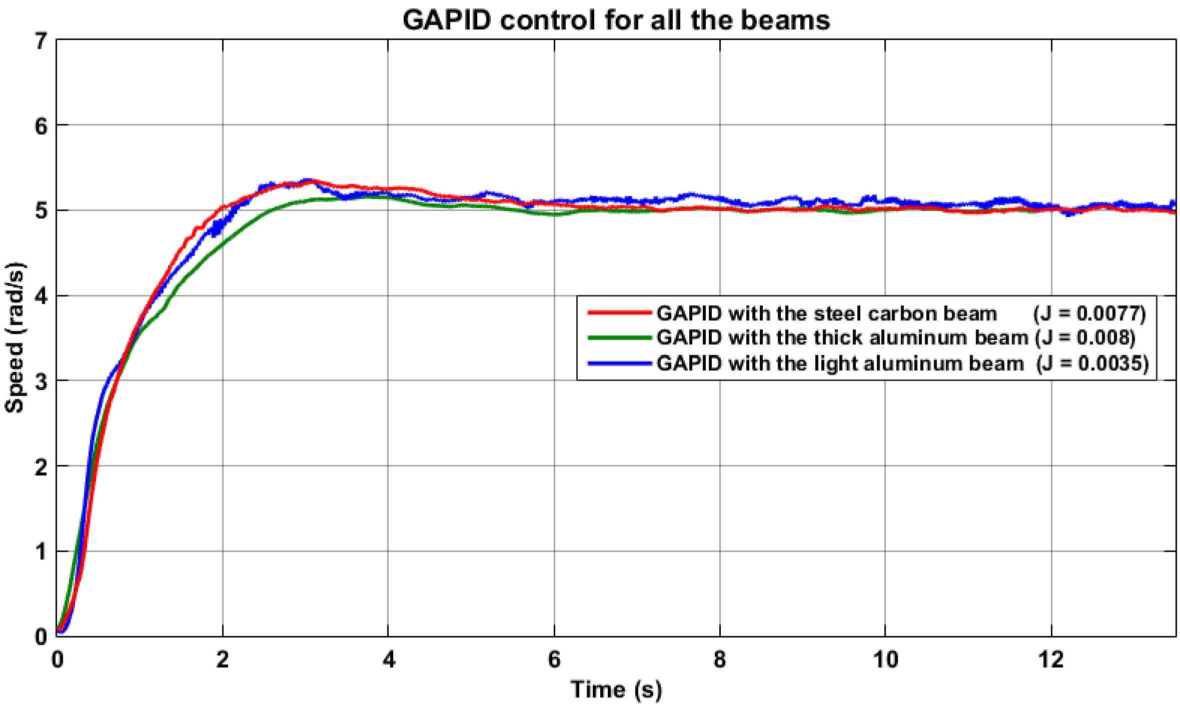
The GAPID parameters found by the PSO are on table (8). The output speed for all three beams with the GAPID control are in Fig. (20).

Table 8 – GAPID parameters for the DC motor

Proportional	Integral	Derivative
$k_{p0} = 0.2$	$k_{i0} = 0.0637$	
$k_{p1} = 100$	$k_{i1} = 0.071$	$k_{d1} = 14$
$q_p = 0.9$	$q_i = 0.9$	$q_d = 0.001$

In Fig. (20), although the slower settling time close to 4 seconds, the overshoot was under 5% for all the three beams. It is noticeable that the output response behave similarly even with the load mass variation in comparison to the performance of the PID.

Graph 20 – GAPID control for all the beams



source: own autorship

5 CONCLUSIONS

The new fitness calculation in equation (18), that imposes a penalty to high overshoot, proved to be an improvement over the fitness in equation (17), reducing the overshoot without sacrificing the response speed (keeping a low settling time) for the study plant while keeping a lower overshoot than the PID for the experimental plant but increasing the settling time.

The bio-inspired optimization algorithms served as efficient methods to tune the GAPID, specially the Particle Swarm Optimization algorithm, which achieved better results than the Genetic algorithm and was used for the other tests as a result. Although, it possibly happened because the limited number of 50 generations was not enough for the GA to converge, as it is averagely slower than the PSO. Only two optimization algorithms were tested. Therefore, other optimization algorithms could possibly achieve even better results.

From the load sweep test, the robustness and adaptive characteristic of the GAPID was validated through simulation. It proved to be efficient even with changes on the load, a characteristic the linear PID does not share, as it loses efficiency as the load changes from the one when it was tuned. Unfortunately, it was not possible to trace the changing behavior of the GAPID parameters with the load sweep, discarding the chance to develop a method to tune the GAPID without the aid of optimization algorithms.

Despite being efficient, the GAPID presented saturation, even if it was for a short time during the transient state and should be considered if it would cause any damage to a physical system. The GAPID improves upon the PID controller, as long as the gain stays near to the projected one. It improves no further otherwise, due to the saturation increase.

In the experimental plant, the GAPID provided a slower response compared to the PID, but it proves its adaptability by maintaining the same output behavior even with the load variation caused by the different beams. The PID, on the other hand, although robust, could not keep the same performance when variation on the plant occurs.

The control method of the GAPID was demonstrated to be applied to different plants with different dynamics. It was possible to apply it to a real system using a PLC and achieve better results than a traditional PID.

Finally, the GAPID is a robust adaptive controller that could improve and optimize the PID controllers used in Industry without causing much nuisance, due to the fact that it uses an already tuned PID and links to it, possibly only requiring some adaptation to the controller programming.

For further studies, new optimization algorithms could be tested to tune the GAPID. This new adaptive controller could also be applied to other plants with different dynamics to analyze the influence of the Gaussian adaptive method.

BIBLIOGRAPHY

AGOSTINHO, A. C. **Controle por modos deslizantes aplicado a sistema de posicionamento dinâmico.** 2009. Tese (Doutorado) — Universidade de São Paulo, 2009. quoted 2 time on pages 24 and 25.

ALEXANDROV, A. G. Frequencial adaptive pid-controller. In: IEEE. **Control Conference (ECC), 1999 European**. [S.l.], 1999. p. 2638–2643. Quoted on page 12.

ÅSTRÖM K. J. ; HÄGGLUND, T. **PID controllers: theory, design, and tuning**. [S.l.]: Instrument society of America Research Triangle Park, NC, 1995. v. 2. Quoted on page 18.

ASTRÖM K. J. ; HÄGGLUND, T. Revisiting the ziegler–nichols step response method for pid control. **Journal of process control**, Elsevier, v. 14, n. 6, p. 635–650, 2004. Quoted on page 18.

ÅSTRÖM K. J. ; WITTENMARK, B. **Adaptive control**. [S.l.]: Courier Corporation, 2013. Quoted on page 12.

ATHERTON D. P ; MAJHI, S. Limitations of pid controllers. In: IEEE. **American Control Conference, 1999. Proceedings of the 1999**. [S.l.], 1999. v. 6, p. 3843–3847. Quoted on page 12.

BIANCHI, L. et al. A survey on metaheuristics for stochastic combinatorial optimization. **Natural Computing**, Springer, v. 8, n. 2, p. 239–287, 2009. Quoted on page 28.

BLONDIN, J. Particle swarm optimization: A tutorial. **from site: http://cs. armstrong. edu/saad/csci8100/pso tutorial. pdf**, 2009. Quoted on page 35.

CASTRO, L. N. De. **Fundamentals of natural computing: basic concepts, algorithms, and applications**. [S.l.]: Chapman and Hall/CRC, 2006. quoted 2 time on pages 36 and 42.

CHEN, Z.; YIN, Y. An new crossover operator for real-coded genetic algorithm with selective breeding based on difference between individuals. In: IEEE. **Natural Computation (ICNC), 2012 Eighth International Conference on**. [S.l.], 2012. p. 644–648. Quoted on page 29.

CLERC, M.; KENNEDY, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. **IEEE transactions on Evolutionary Computation**, IEEE, v. 6, n. 1, p. 58–73, 2002. quoted 2 time on pages 35 and 36.

COLEY, D. A. **An introduction to genetic algorithms for scientists and engineers**. [S.l.]: World Scientific Publishing Company, 1999. Quoted on page 28.

CORNETTI, G. P. Implementação e sintonia de um controlador clássico industrial PID: estudo de caso na indústria química. In: . [S.l.]: Universidade Estadual Paulista (UNESP), 2014. quoted 2 time on pages 18 and 19.

DORF, R. C.; BISHOP, R. H. Sistemas de controle codernos, 8ª edição. **Editora LTC, Rio de Janeiro**, 2001. Quoted on page 20.

EBERHART, R. C.; SHI, Y. Comparing inertia weights and constriction factors in particle swarm optimization. In: IEEE. **Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)**. [S.l.], 2000. v. 1, p. 84–88. Quoted on page 42.

GOLDBERG, D. **The design of innovation (Genetic algorithms and evolutionary computation)**. [S.l.]: Springer New York, 2002. Quoted on page 31.

GWIAZDA, T. D. **Crossover for single-objective numerical optimization problems**. [S.l.]: Tomasz Gwiazda, 2006. v. 1. quoted 2 time on pages 32 and 33.

INSTRUMENTS, NI National. **CompactRIO systems**. 2019. Http://www.ni.com/pt-br/shop/compactrio.html. Acessed on 05/22/2019. Quoted on page 45.

IPPOLITI, E. **Heuristic reasoning**. [S.l.]: Springer, 2014. v. 16. Quoted on page 28.

JANZEN, F. C. et al. Controle do movimento de rastreamento de uma estrutura flexivel sujeita a amortecimento viscoso. In: **Congresso Nacional de Engenharia Mecânica CONEM**. [S.l.: s.n.], 2014. Quoted on page 21.

JIMENEZ-URIBE, Á et al. Comparative study of a pid and pd control bounded by hyperbolic tangent function in robot 3 dof. In: IEEE. **Mechatronics, Electronics and Automotive Engineering (ICMEAE), 2015 International Conference on**. [S.l.], 2015. p. 199–204. Quoted on page 12.

JR, I. Fister et al. A brief review of nature-inspired algorithms for optimization. **arXiv preprint arXiv:1307.4186**, 2013. Quoted on page 13.

JULIAN, P et al. Output discrete feedback control of a dc-dc buck converter. In: IEEE. **Industrial Electronics, 1997. ISIE'97., Proceedings of the IEEE International Symposium on**. [S.l.], 1997. p. 925–930. Quoted on page 16.

KAPUR, R. Review of nature inspired algorithms in cloud computing. In: IEEE. **Computing, Communication & Automation (ICCCA), 2015 International Conference on**. [S.l.], 2015. p. 589–594. Quoted on page 13.

KASTER, M. S. et al. Análise do uso de controle pid não linear aplicado a conversor buck. In: **Proceedings of the X Conferência Brasileira de Dinâmica, Controle e Aplicações**. [S.l.: s.n.], 2011. Quoted on page 13.

____. Gaussian adaptive pid control optimized via genetic algorithm applied to a step-down dc-dc converter. In: **Proceeding of the 12th IEEE International Conference on Industry Applications (INDUSCON 2016)**. [S.l.: s.n.], 2016. Quoted on page 20.

KENNEDY, J.; MENDES, R. Population structure and particle swarm performance. In: IEEE. **Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)**. [S.l.], 2002. v. 2, p. 1671–1676. Quoted on page 36.

KENNEDY, R. J. and eberhart, particle swarm optimization. In: **Proceedings of IEEE International Conference on Neural Networks IV, pages**. [S.l.: s.n.], 1995. v. 1000. quoted 2 time on pages 34 and 42.

KHAN, T. A. et al. Modeling of a standard particle swarm optimization algorithm in matlab by different benchmarks. In: IEEE. **Second International Conference on the Innovative Computing Technology (INTECH 2012)**. [S.l.], 2012. p. 271–274. Quoted on page 42.

LIOU, W.; YEH, M.; KUO, Y. A high efficiency dual-mode buck converter ic for portable applications. **IEEE Transactions on Power Electronics**, IEEE, v. 23, n. 2, p. 667–677, 2008. Quoted on page 16.

LIPOWSKI, A.; LIPOWSKA, D. Roulette-wheel selection via stochastic acceptance. **Physica A: Statistical Mechanics and its Applications**, Elsevier, v. 391, n. 6, p. 2193–2196, 2012. quoted 2 time on pages 30 and 31.

LU, K. et al. Comparison of binary coded genetic algorithms with different selection strategies for continuous optimization problems. In: IEEE. **Chinese Automation Congress (CAC), 2013**. [S.l.], 2013. p. 364–368. Quoted on page 30.

MAITY, A. et al. Design of a 20 mhz dc-dc buck converter with 84 percent efficiency for portable applications. In: IEEE. **VLSI Design (VLSI Design), 2011 24th International Conference on**. [S.l.], 2011. p. 316–321. Quoted on page 16.

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 2 time on pages 31 and 32.

MITCHELL, M. **An introduction to genetic algorithms**. [S.l.]: MIT press, 1998. Quoted on page 29.

NISE, N. S. **Control System Engineering**. [S.l.]: John Wiley & Sons, 2007. Quoted on page 12.

OGATA, K. **Modern control engineering**. [S.l.]: Prentice hall, 2010. v. 5. Quoted on page 12.

OLIVEIRA, R. M. **Aplicação de controle PID não linear com ganhos baseados em perfil gaussiano aplicado a um conversor Buck**. 2014. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2014. Quoted on page 24.

PEZZELLA, F.; MORGANTI, G.; CIASCHETTI, G. A genetic algorithm for the flexible job-shop scheduling problem. **Computers & Operations Research**, Elsevier, v. 35, n. 10, p. 3202–3212, 2008. Quoted on page 34.

POLOLU. **Pololu Dual VNH5019 Motor Driver Shield for Arduino**. 2019. Https://www.pololu.com/product/2507. Acessed on 05/17/2019. Quoted on page 44.

PUCHTA, E. D. P. et al. Gaussian adaptive pid control optimized via genetic algorithm applied to a step-down dc-dc converter. In: IEEE. **Industry Applications (INDUSCON), 2016 12th IEEE International Conference on**. [S.l.], 2016. p. 1–6. Quoted on page 36.

PUCHTA, E. D. P. et al. **Controle PID gaussiano com otimização dos parâmetros das funções gaussianas usando algoritmo genético e PSO**. 2016. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2016. quoted 2 time on pages 13 and 26.

RIFAI, K. El. Nonlinearly parameterized adaptive pid control for parallel and series realizations. In: IEEE. **American Control Conference, 2009. ACC'09.** [S.l.], 2009. p. 5150–5155. Quoted on page 12.

RODRIGUEZ, M. et al. A multiple-input digitally controlled buck converter for envelope tracking applications in radiofrequency power amplifiers. **IEEE Transactions on Power Electronics**, IEEE, v. 25, n. 2, p. 369–381, 2010. Quoted on page 16.

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 18.

SCHMITT, L. M. Theory of genetic algorithms. **Theoretical Computer Science**, Elsevier, v. 259, n. 1-2, p. 1–61, 2001. Quoted on page 30.

SHARMA, A.; SINHA, M. Influence of crossover and mutation on the behavior of genetic algorithms in mobile ad-hoc networks. In: IEEE. **Computing for Sustainable Global Development (INDIACom), 2014 International Conference on**. [S.l.], 2014. p. 895–899. Quoted on page 32.

SHI, Y.; EBERHART, R. A modified particle swarm optimizer. In: IEEE. **1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)**. [S.l.], 1998. p. 69–73. Quoted on page 35.

SHI, Y.; EBERHART, R. C. Parameter selection in particle swarm optimization. In: SPRINGER. **International conference on evolutionary programming**. [S.l.], 1998. p. 591–600. Quoted on page 42.

SHI, Y.; EBERHART, R. C. Parameter selection in particle swarm optimization. In: SPRINGER. **International conference on evolutionary programming**. [S.l.], 1998. p. 591–600. Quoted on page 54.

SINGH, B.; KUMAR, V. A real time application of model reference adaptive pid controller for magnetic levitation system. In: IEEE. **Power, Communication and Information Technology Conference (PCITC), 2015 IEEE**. [S.l.], 2015. p. 583–588. Quoted on page 12.

SUNG, S. W.; LEE, I. Limitations and countermeasures of pid controllers. **Industrial & engineering chemistry research**, ACS Publications, v. 35, n. 8, p. 2596–2610, 1996. Quoted on page 12.

SWIECH, M. C. S.; OROSKI, E.; ARRUDA, L. V. R. D. Sintonia de controladores pid em colunas de destilação através de algoritmos genéticos. In: **3rd Congresso Brasileiro de P &D em Petróleo e Gás (submetido), Salvador, BA**. [S.l.: s.n.], 2005. quoted 2 time on pages 13 and 19.

TAY, T.; MAREELS, I.; MOORE, J. B. **High Performance Control (Systems & Control: Foundations & Applications)**. [S.l.]: Birkhäuser, 1997. Quoted on page 26.

UTKIN, V. I. Sliding modes and their applications in variable structure systems. **Mir, Moscow**, 1978. Quoted on page 24.

VESTERSTROM, J.; THOMSEN, R. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: IEEE. **Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)**. [S.l.], 2004. v. 2, p. 1980–1987. Quoted on page 42.

WANG, R.; ZHANG, Z. Fuzzy adaptive pid hybrid control for airborne platform mounted on muav. In: IEEE. **Information Technology, Networking, Electronic and Automation Control Conference, IEEE**. [S.l.], 2016. p. 737–741. Quoted on page 12.

WILSON, E. O. Sociobiology: the new synthesis. **Cambridge, MA: Belknap**, 1975. Quoted on page 34.

XIAO, J. et al. Robot force position control without impact: method and application. In: IEEE. **Machine Learning and Cybernetics (ICMLC), 2010 International Conference on**. [S.l.], 2010. v. 2, p. 965–968. Quoted on page 12.

____. A 4-/spl mu/a quiescent-current dual-mode digitally controlled buck converter ic for cellular phone applications. **IEEE Journal of Solid-State Circuits**, IEEE, v. 39, n. 12, p. 2342–2348, 2004. Quoted on page 16.

XIONG, A.; FAN, Y. Application of a pid controller using mrac techniques for control of the dc electromotor drive. In: IEEE. **Mechatronics and Automation, 2007. ICMA 2007. International Conference on**. [S.l.], 2007. p. 2616–2621. Quoted on page 12.

A genetic algorithm with multi-step crossover for job-shop scheduling problems. [S.l.]: IET, 1995. 960-969 p. ISSN 0302-9743. Quoted on page 32.

YAN, W. et al. Fixed-frequency boundary control of buck converter with second-order switching surface. **IEEE Transactions on Power Electronics**, v. 24, n. 9, p. 2193–2201, Sept 2009. ISSN 0885-8993. Quoted on page 25.