

# Optimal tuning of a proportional controller for DC motor position control via Fibonacci Search Method

DENIS MOSCONI<sup>1</sup>, MAIRA MARTINS DA SILVA<sup>2</sup>, ADRIANO ALMEIDA GONÇALVES SIQUEIRA<sup>2</sup>

<sup>1</sup>Federal Institute of São Paulo

<sup>2</sup>University of São Paulo

<denis.mosconi@ifsp.edu.br>, <mairams@sc.usp.br>, <siqueira@sc.usp.br>

DOI: <http://dx.doi.org/10.21439/jme.v4i1.88>

**Received:** 08 Feb. 2021. **Accepted:** 09 Apr. 2021

**Abstract.** One of the advantages of using DC motors is the ease of controlling their position and speed by manipulating the input voltage. To this, feedback PID-based controllers can be used. However, adjusting these controllers can be challenging and require some reasonable effort from the controller designer. This work proposes an algorithm based on the Fibonacci Search Method to determine the optimal gain of a proportional controller applied to the position control of a DC motor. The project specifications were minimum settling time with null overshoot. The results obtained showed that the proposed method is valid for determining the  $K_p$  gain according to the plant and the conditions involved. The proposed method was compared with other optimization techniques such as Golden Section, Quasi-Newton and Grey Wolf Optimization, standing out for its simplicity of implementation, low number of iterations and fast convergence.

**Keywords:** Optimization. Feedback control. Direct current motor.

## 1 Introduction

Direct Current motors (DC motors) are electrical machines capable of convert direct current and voltage into movement. The first DC motor was developed in 1832 by the British electrician and scientific lecturer William Sturgeon (GEE; JAMES, 2004). From then on, this type of engine started to be widely used, being present in the most varied areas, such as robotics, medicine, industry, military, toys, and domestic devices. In 2016, the global electric DC motors market size was valued at USD 20.196 million and its rising; that is expected that in 2025 this value will reach USD 35.6 billion (GVR, 2017).

Among the several advantages of using a DC motor, we highlight the ease of controlling its speed and position, by manipulating the input voltage. To perform these controls, the well-known PID-based controllers (such as P, PD, PI or even PID) can be applied. Despite this type of controllers are very well known and largely applied, they have a challenge: to determine the

optimal gains that ensure its good performance and efficacy, according to the project requirements.

Several methods were proposed to tune the PID-based controllers with optimal gains, such as Ziegler-Nichols (ZIEGLER; NICHOLS, 1942), Root Locus (EVANS, 1948), artificial intelligence based (FLORES-MORAN; YANEZ-PAZMINO; BARZOLA-MONTESES, 2018) and optimization techniques (RAMYA; JADHAV; PAWAR, 2020). Although several tuning methods exist, the Ziegler-Nichols and trial-and-error are the more applied methods because of their relative ease of application. These methods take time and effort on the part of the controller designer and may not be applicable to certain plants, in addition to not always providing optimal gains as expected.

In this work we apply the Fibonacci Search Method (FSM) to determine the optimal gain of a feedback proportional controller applied to the position control of a DC motor. In this case, *optimal gain* means the best  $K_p$  gain that makes the system respond to a step input as

quickly as possible (shortest settling time) without any overshoot, that is, the gain that ensures the system to be critically damped.

Both the optimization technique and the proportional control were chosen due to their low computational cost and simplicity of implementation, so that the use of these methods can be used in any control unit, such as microcontrollers, which allows this work be extended to hardware-in-the-loop systems.

A hypothesis is formulated for this work: through the use of FSM, the  $K_p$  gain that satisfies the project conditions can be determined quickly and with low computational cost.

This paper is organized as follows: Section 2 presents a revision about related works, Section 3 contains the dynamics modeling of DC motor, a stability analysis is performed in Section 4, the Fibonacci Search Method and its application in this work are presented in Section 5, an analytical determination of  $K_p$  is performed in Section 6, the results obtained with the application of FSM are presented in Section 7, the conclusion is in Section 8 and the Section 9 presents ideas for future works correlated.

## 2 Bibliographic Review

Several works related to tuning feedback PID-based controllers applied to DC motor have been developed. In this section, we will discuss some of the more recent ones.

A tuning method based on the Fibonacci Search Method was proposed by Horla e Sadalla (2020) to determine the optimal gains of a Fractional-order Proportional-Integral (FOPI) controller. With this method, the researchers could determine more than one controller parameter, focusing in to minimize a cost function took as the integrated absolute tracking error. The results obtained proved that the method is applicable and has the advantage of not requiring the plant model to determine the controller gains.

Ramya, Jadhav e Pawar (2020) used Particle Swarm Optimization (PSO) to tuning a PID controller for control of speed of Permanent Magnet Brush Less DC motor. This type of motor has a large application area such as aeronautics, medical, robots, chemical and industrial automation. When compared to the Ziegler-Nichols and Genetic Algorithm methods, the PID tuned by PSO presented a better efficiency with faster settling time and negligible overshoot.

A recently developed optimization method, the Sine

Cosine Algorithm (MIRJALILI, 2016), was used by Thakur et al. (2020) to tuning a PID controller of a DC servo-motor system. This tuning was performed by minimizing the integral-square-error. Although the method used manages to stabilize the motor in a steady state, a 20% overshoot exists for the step input, which may be undesirable in some applications. In addition, the authors did not perform any stability analysis or comparison with other optimization methods for tuning.

Kouassi et al. (2019) resorted to Ant Colony Optimization (ACO) to tuning PID controller for DC motor speed control. With six iterations the ACO determined the optimal gains that made the system present a null overshoot and a settling time of 0.29 seconds, against 6.96% overshoot and 0.38 seconds of settling time with the gains determined by the Ziegler-Nichols method.

Atom Search Optimization (ASO) and a variation of it, Chaotic Atom Search Optimization (ChASO) were used by Hekimoglu (2019) to tuning a fractional-order proportional-integral-derivative (FOPID) controller, applied to speed control of DC motor. The optimal gains were determined after eight iterations. Both ASO-FOPID and ChASO-FOPID controller promoted null overshoot and 0.06 and 0.04 seconds settling time, respectively. The proposed approach was compared to different optimal tuning methods and controllers, and the ChASO-FOPID presented the best results.

A comparison study between tuning PID controller using fuzzy and Genetic Algorithm (GA) was performed by Flores-Moran, Yanez-Pazmino e Barzola-Monteses (2018). The controller was used to control position of DC motor. Both methods resulted in a 5% overshoot, but the PID controller tuned by fuzzy presented a fast action with a settling time of 0.4 s while the controller tuned by GA presented a slow action with a settling time of 0.78 s.

Achanta e Pamula (2017) compared the Particle Swarm Optimization (PSO) and Jaya Optimization Algorithm (JOA) for tuning the PID controller applied to a DC motor speed control. With PSO the settling time and overshoot were 0.40s and 0.0063% respectively while with JOA, the values obtained were 0.52 s for settling time and 2.28% for overshoot which leads to the conclusion that PSO is better than the JOA for this application.

A Novel Fuzzy Single Neuron PID (NFSNPID) controller was proposed by Ghany, Shamseldin e Ghany (2017). To design this controller, ANN and fuzzy techniques were used, being the design divided in two parts:

the first part uses Genetic Algorithm to obtain optimal parameters for SNPID (Single Neuron PID) controller and the second part deals with the design of fuzzy logic control that update the weights of SNPID control on-line. This controller was applied to a DC motor speed control and presented the best results when compared to the SNPID and CFSNPID (Conventional Fuzzy Single Neuron PID) control. With this approach the settling time and overshoot obtained were 0.016 s and 0.0096%, respectively.

Many of the works discussed deal with the design of PID controllers. In this work a single proportional controller is applied, due to its simplicity of implementation and the type of the controlled plant.

### 3 Dynamical Model of DC motor

In order to determine the transfer function of the DC motor, a mathematical modeling of its dynamics is performed here. This modeling is based on the schematic depicted by the Figure 1.

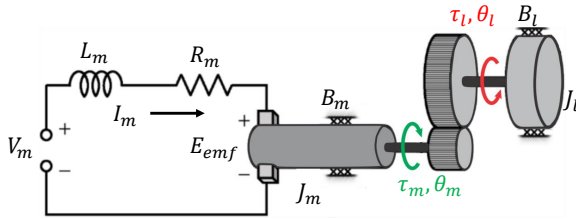


Figure 1: DC motor schematic

Applying the Kirchhoff's voltage law to the electrical circuit, the Equation (1) is obtained.

$$V_m - R_m I_m - L_m \frac{dI_m}{dt} - E_{emf} = 0 \quad (1)$$

$V_m$  is the voltage applied to the motor,  $I_m$  is the current,  $R_m$  is the winding resistance,  $L_m$  is the characteristic inductance of the rotor and  $E_{emf}$  is the induced counter-electromotive force, proportional to the angular velocity of the rotor. Because  $L_m \ll R_m$ , the inductance can be neglected. Then, the current in the winding can be expressed by Equation (2).

$$I_m = \frac{V_m - K_m \dot{\theta}}{R_m} \quad (2)$$

where  $K_m$  is the counter-electromotive constant and  $\dot{\theta}$  is the angular velocity of the rotor.

Applying the second Newton's law to the mechanical system the Equation (3) is obtained.

$$J_l \ddot{\theta}_l = K_g \eta_g \tau_m - K_g \eta_g J_m \ddot{\theta}_m - K_g b_m \dot{\theta}_m - b_l \dot{\theta}_l \quad (3)$$

$J_l$  and  $J_m$  are the load and rotor inertia, respectively,  $K_g$  is the gear ratio,  $\eta_g$  is the gear efficiency,  $\tau_m$  is the torque provided by the motor and  $b_l$  and  $b_m$  are the load and rotor damping coefficient, respectively.

Using the transformations  $\tau_m = \eta_m K_t I_m$  (being  $\eta_m$  the motor efficiency and  $K_t$  the torque constant of the motor),  $\theta_m = K_g \theta_l$  and  $b_{eq} = K_g^2 b_m + b_l$ , Equation (3) can be rewritten as:

$$J_l \ddot{\theta}_l + \eta_g K_g^2 J_m \ddot{\theta}_l + B_{eq} \dot{\theta}_l = \eta_g \eta_m K_g K_t I_m \quad (4)$$

Finally taking  $J_{eq} = J_l + \eta_g K_g^2 J_m$ , combining the equations (4) and (2) and applying the Laplace transform (all the initial conditions are null), the transfer function that establishes the relationship between the angular position of the load coupled to the axle ( $\theta_l$ ) and the voltage applied to the motor ( $V_m$ ) is given by:

$$\frac{\theta_l(s)}{V_m(s)} = \frac{\eta_g \eta_m K_g K_t}{J_{eq} R_m s^2 + (B_{eq} R_m + \eta_g \eta_m K_g^2 K_t K_m) s} \quad (5)$$

Substituting the parameter values (these values are based on the Quanser Rotary Servo Base Unit SRV02 and shown in the appendix), the transfer function becomes:

$$\frac{\theta_l(s)}{V_m(s)} = \frac{3673.07}{s^2 + 36.4s} \quad (6)$$

### 4 Lyapunov Stability Analysis

The main objective of this work is to find the  $K_p$  gain of the feedback control (Fig. 2) that satisfies the desired conditions. For this, is necessary to perform a stability analysis, in order to verify what are the possible values that  $K_p$  can assume without leading the system to instability. With this analysis, the optimization constraints are determined.

Taking  $\theta^d(s) = 0$ ,  $D(s) = 0$  and  $N(s) = 0$ , the transfer function relating the output  $\theta(s)$  and the error  $e(s)$  is given by Equation (7).

$$\frac{\theta(s)}{e(s)} = \frac{3673.07 K_p}{s(s + 36.4)} \quad (7)$$

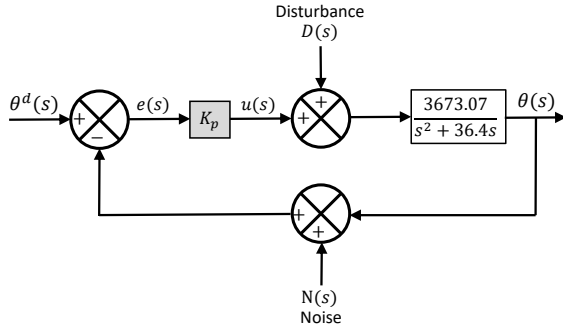


Figure 2: Proportional feedback control

Being the position error the difference between the desired and real positions ( $e(s) = \theta^d(s) - \theta(s)$ ), Equation (7) can be rewritten as:

$$s^2 e(s) + 26.4s e(s) + 3673.07 K_p e(s) = 0 \quad (8)$$

Applying the inverse Laplace transform to Equation (8), the dynamics of the error is then expressed as:

$$\ddot{e} + 36.4\dot{e} + 3673.07 K_p e = 0 \quad (9)$$

Expressing Equation (9) in state variables:

$$\begin{aligned} x_1 &= e \\ x_2 &= \dot{e} = \dot{x}_1 \\ \dot{x}_2 &= -3673.07 K_p x_1 - 36.4 x_2 \end{aligned} \quad (10)$$

Choosing a matrix  $P$ :

$$\begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix} \quad p_1, p_2 > 0 \quad (11)$$

Defining the Lyapunov candidate function as:

$$V(x) = \frac{1}{2} x^T P x = \frac{1}{2} p_1 x_1^2 + \frac{1}{2} p_2 x_2^2 \quad (12)$$

The time derivative of Equation (12) is:

$$\begin{aligned} \dot{V}(x) &= p_1 x_1 \dot{x}_1 + p_2 x_2 \dot{x}_2 \\ &= p_1 x_1 x_2 - 3673.07 K_p p_2 x_1 x_2 - 36.4 p_2 x_2^2 \end{aligned} \quad (13)$$

Expressing Equation (13) in matrix form:

$$\begin{aligned} \dot{V}(x) &= x^T \begin{bmatrix} 0 & \frac{p_1 - p_2 3673.07 K_p}{2} \\ \frac{p_1 - p_2 3673.07 K_p}{2} & -36.4 p_2 \end{bmatrix} x \\ &= -x^T N x \end{aligned} \quad (14)$$

Equation (14) will always be negative semidefinite if  $-N$  be negative semidefinite. The matrix  $-N$  will be negative semidefinite if all principal minors of  $N$  are positive or null, that is:

$$36.4 p_2 \geq 0 \quad (15)$$

$$\frac{-(p_1 - p_2 3673.07 K_p)^2}{4} \geq 0 \quad (16)$$

As  $p_2 > 0$ , Equation (15) is always satisfied. Otherwise, Equation (16) can only satisfy the equality, which results in

$$p_1 = p_2 3673.07 K_p \quad (17)$$

To satisfy Equation (17), the gain  $K_p$  must be positive, thus, the derivative of the Lyapunov candidate function becomes  $\dot{V}(x) = -36.4 p_2 x_2^2$  which is always negative semidefinite. Then, to ensure the system stability, it is necessary to have  $K_p > 0$ , and this is the optimization constrain.

## 5 Fibonacci Optimization Method and the Control Tuning

Given an objective function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  of the independent variables  $\mathbf{x}$ , the optimization problem is to find a vector  $\mathbf{x}^*$  subject to some constrain and that minimizes (or maximizes)  $f$ . In this work, the purpose is to minimize a function  $f$ , then, sometimes the words *optimization* and *minimization* will be used indistinctly.

The Fibonacci Search Method (CHONG; ZAK, 2012; RAO, 2009) is a tool to solve one-dimensional optimization problems with unimodal objective functions. Such a method works as follows: given an initial closed interval, say  $[L_i, L_s]$ , it seeks to find a new closed interval  $[L'_i, L'_s] \subset [L_i, L_s]$  that is as small as possible and that contains the minimizer  $x^*$ , with as few iterations as possible. For this,  $f$  is evaluated at two intermediate points of  $[L_i, L_s]$ ,  $a$  and  $b$ , that are determined as:

<sup>1</sup>Vector quantities are in bold and scalar quantities are not.

$$\begin{aligned} a &= L_i + \rho(L_s - L_i) \\ b &= L_s + (1 - \rho)(L_s - L_i) \end{aligned} \quad (18)$$

Then, if  $f(a) < f(b)$  the initial interval becomes  $[L_i, b]$ , otherwise if  $f(a) > f(b)$  the initial interval becomes  $[a, L_s]$ . After this evaluation and redefinition of the search interval,  $\rho$  is updated and the process restarts. This algorithm repeats until the maximum number of iterations is reached. At each iteration  $k$ ,  $\rho_k$  is updated as:

$$\rho_k = 1 - \frac{F_{N-k+1}}{F_{N-k+2}} - \epsilon \quad (19)$$

where  $F_k$  are the elements of the Fibonacci sequence. This sequence is defined as follows. First, let  $F_{-1} = 0$  and  $F_0 = 1$  by convention. Then, for  $k \geq 0$ :

$$F_{k+1} = F_k + F_{k-1} \quad (20)$$

Some elements in the Fibonacci sequence are:

$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$
1	2	3	5	8	13	21	34

The  $\epsilon$  in Equation (19) is due to an anomaly in the last iteration of the Fibonacci Search Method, because:

$$\rho_N = 1 - \frac{F_1}{F_2} = \frac{1}{2} \quad (21)$$

What makes the boundaries of the new interval coincide, which is undesirable. Then, in the last iteration  $\rho_N = \frac{1}{2} - \epsilon$ , where  $\epsilon$  is a small number ( $10^{-3}$  in this work). In other iterations,  $\epsilon = 0$ . A flowchart of the Fibonacci Search Method used in this work can be seen in Figure 3, the algorithm was developed in a way to reduce computational cost. The maximum number of iterations was fixed as  $N = 6$  and the boundaries of the initial interval are  $[0.001, 1]$ .

The optimization problem of this work is presented as:

$$\begin{aligned} \text{minimize } f(K_p) &= \alpha M_p + \beta t_s \\ \text{subject to } K_p &> 0 \end{aligned} \quad (22)$$

where  $\alpha$  and  $\beta$  are weights (unitary in this work),  $M_p$  and  $t_s$  are the overshoot and settling time, respectively, determined by equations (23) and (24).

$$M_p = e^{\frac{-\xi\pi}{\sqrt{1-\xi^2}}} \quad (23)$$

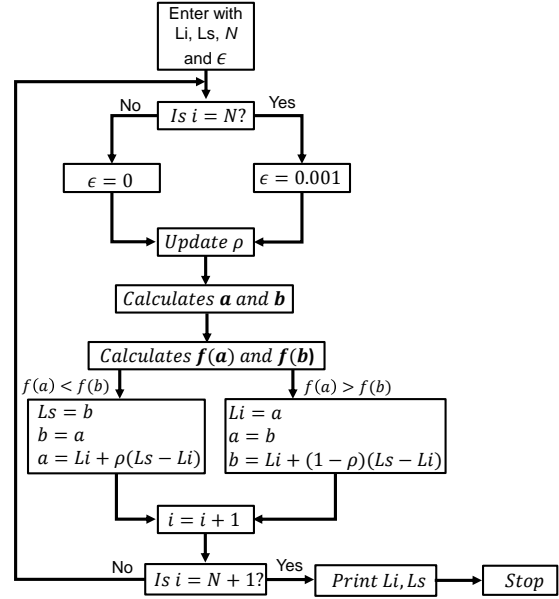


Figure 3: Fibonacci Search Method flowchart

$$t_s = \frac{4}{\xi\omega_n} \quad (24)$$

In this case,  $\xi$  and  $\omega_n$  are:

$$\omega_n = \sqrt{3673.07K_p} \quad (25)$$

$$\xi = \frac{\sqrt{3673.07K_p}}{201.82K_p} \quad (26)$$

After obtained the new interval  $[L'_i, L'_s]$ , the optimum  $K_p$  gain is determined by the center of mass of this new interval:

$$K_p = \frac{1}{m_i + m_s} (L'_i m_i + L'_s m_s) \quad (27)$$

where  $m_i = f^{-1}(L'_i)$  and  $m_s = f^{-1}(L'_s)$ .

The center of mass is used to determine the best  $K_p$  gain in the new interval obtained with the Fibonacci Search Method, this approach is better than using the arithmetic mean over the interval.

The algorithm was developed in MATLAB® R2017a and the executions were carried out on a computer with Intel®Core™i7-5500 2.40 GHz processor, 8.00 GB of RAM, 2.00 GB dedicated video card, Windows 10 Home Single Language 64 bits.

## 6 Analytical Determination of $K_p$

The closed-loop transfer function of the system is given by:

$$\frac{\theta(s)}{\theta^d(s)} = \frac{3673.07K_p}{s^2 + 36.4s + 3673.07K_p} \quad (28)$$

The second-order system transfer function in the standard form is:

$$\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (29)$$

In this work, the  $K_p$  gain must ensure that both overshoot and settling time will be the minimum as possible ( $\alpha$  and  $\beta$  where chosen as unitary). To this, it is necessary to have  $\xi = 1$  which results in  $K_p = 0.0902$ . Then, in this case, is expected that the optimization find this value for  $K_p$ , proving its effectiveness.

## 7 Results and Discussions

The value of gain found through the Fibonacci Search Method is  $K_p = 0.1025$  which is 0.0123 greater than the one determined by the analytical method (see section 6). This gain was determined after 6 iterations, which is obvious, since  $N = 6$  (see Figure 3) and the time spent in the execution was 0.45s.

Analyzing Figure 4, is possible to see a fast convergence of the overshoot ( $M_p$ ), while a small variation of the settling time ( $t_s$ ) is noticed. This constancy of the settling time is due to the nature of the closed-loop system, then to promote considerable changes in  $t_s$ , another type of controller must be used (e.g. proportional-derivative).

The interval of possible optimal gains values also was quickly reduced, as can be seen in the Figure 5. The initial value was  $[0.001, 1]$  and the final one was  $[0.096, 0.1437]$ , being their span 0.999 and 0.047, respectively. The final interval has a span that is 95% less than the initial one.

The step response of the system can be analyzed for the gains obtained through the Fibonacci Search Method and the Analytical Method, looking at Figure 6. In both, the reference value is reached in the steady state and there is no overshoot, however with the gain determined through FSM, the system converges more quickly. The steady state error equal zero for a step input was expected since the system is type 1.

A comparison between the results obtained with FSM and analytical methods is provided by Table 1.

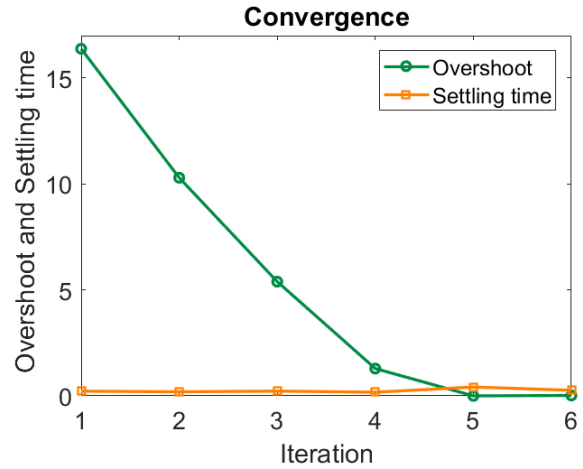


Figure 4: Overshoot and Settling Time convergences along the iterations.

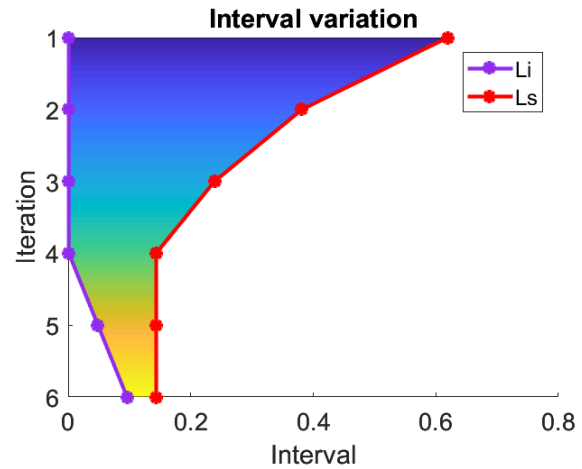
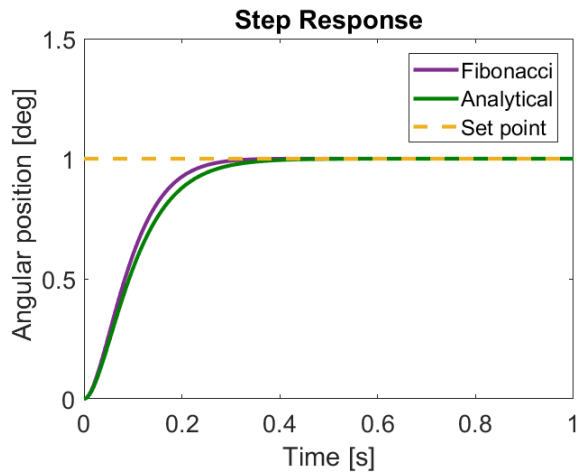


Figure 5: Overshoot and Settling Time convergences along the iterations.

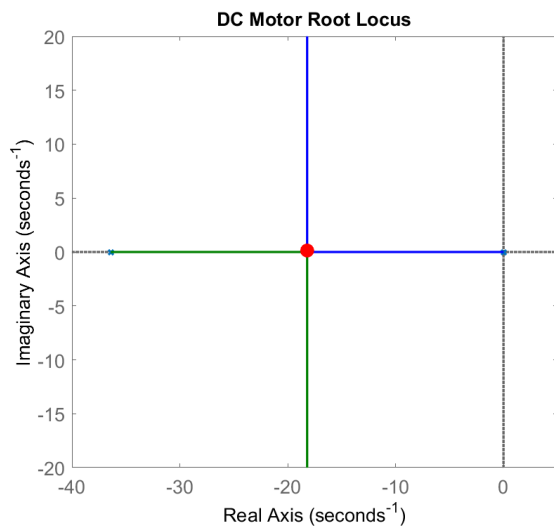
Table 1: Comparison between the results obtained with Analytical and Fibonacci Search methods

	Fibonacci	Analytical
$K_p$	0.1025	0.0902
Overshoot (%)	0.0196	0
Settling Time (s)	0.26	0.32



**Figure 6:** Overshoot and Settling Time convergences along the iterations.

The MATLAB® function *rlocfind* was used to determine the  $K_p$  gain using root locus. The value obtained was  $K_p = 0.0902$ , the same obtained analytically. A plot of the root locus can be seen in Figure 7.



**Figure 7:** Root locus of the system. The red dot depicts the point that was used to determine  $K_p$ .

Although the three methods (analytical, root locus and FSM) provide good values of  $K_p$  the Fibonacci Search Method has the advantage of to be fast. Furthermore, only the motor parameters and the values of  $\alpha$  and  $\beta$  must be informed to the FSM, while the analyti-

cal method requires the user to solve equations and the root locus demands greater knowledge of control theory.

The Fibonacci Search Method (FSM) was compared to Golden Section (GS), Quasi-Newton (Quasi) and Grey Wolf Optimizer (GWO) (MIRJALILI; MIRJALILI; LEWIS, 2014) methods, to determination of  $K_p$ . The results obtained with each method are presented in Table 2. In order to make a fair comparison between the mentioned optimization methods, the computer was restarted and only MATLAB was running during the execution of the algorithms.

**Table 2:** Comparison between the results obtained with different optimization methods

	FSM	GS	Quasi	GWO
$K_p$	0.102	0.096	0.098	0.100
$M_p$ (%)	0.019	0	0	0
$t_s$ (s)	0.26	0.28	0.28	0.27
Exec. time* (s)	0.45	0.50	0.97	9.30
Iterations	6	6	6	51
Obj. function**	12	12	30	306

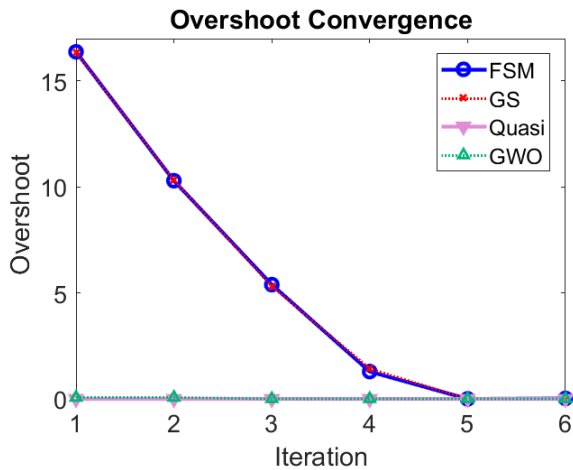
\*Execution time.

\*\*Number of times the objective function was calculated during optimization.

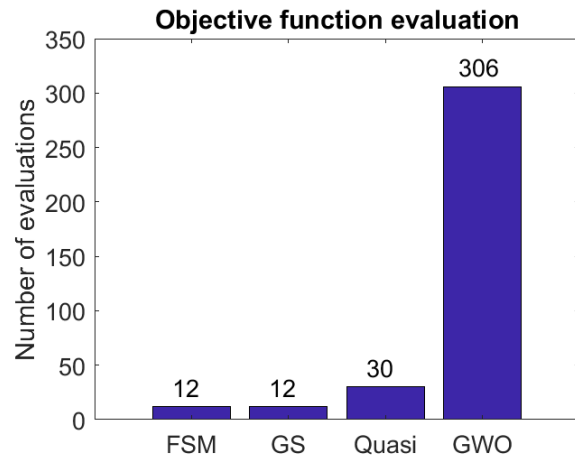
The overshoot convergence for each method is presented in Figure 8. The settling time was not plotted because its low variation due to the nature of the closed-loop system.

The results obtained with the GS method are very close to the ones obtained with FSM. This is due to the fact that both algorithms are very similar. The  $K_p$  gain obtained with GS is closer to the one calculated by the analytical method, however the settling time and the time took by the simulation are slightly larger than the ones from FSM. In both methods, only 6 iterations were needed to achieve the optimal gain and the objective function was analyzed only 12 times during the execution of the algorithms.

Despite the quick convergence, the Quasi-Newton method has a major limitation: it is sensible to the arbitrary initial value of  $K_p$ . If a bad choice of this initial value is made, the algorithm does not converge and the optimal gain is not found. When compared to FSM, the Quasi algorithm took twice as long to be executed, in addition the objective function was calculated 30 times and the results obtained were not extremely better than those obtained with FSM.



**Figure 8:** Overshoot convergence for each optimization method tested



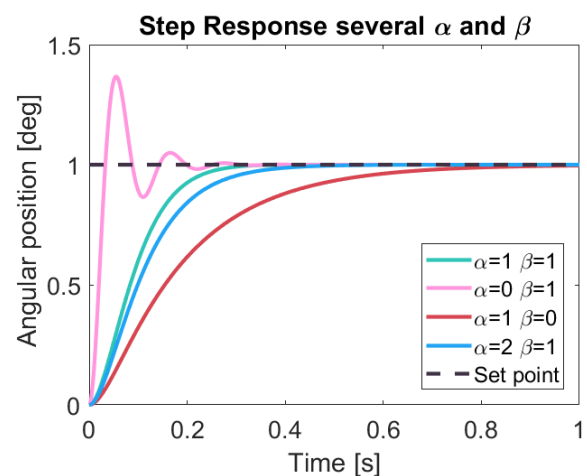
**Figure 9:** Number of times the objective function was calculated for each optimization method

The meta-heuristic optimizer GWO also showed a fast convergence. However it took much more time to be executed, in addition to requiring much more iterations, which resulted in a large amount of calculation of the objective function.

If an autotuning system is developed using an optimization method, and the plant replaces the cost function, then for each time the objective function would be calculated, the plant is submitted to a test entry (e.g. step input) in order to have its response to stimulus evaluated. So, the number of times the objective function is calculated is important in determine how optimization method will be used to develop this autotuning system. Comparing the four methods tested, we can affirm that FSM and GM are good choices for this purpose (Fig. 9).

A further test was carried out, varying the values of  $\alpha$  and  $\beta$  in Equation (22). The results obtained with this test are shown in Figure 10. As expected, if the reduction of overshoot is prioritized (i.e. the  $\alpha$  value is increased over  $\beta$ ) more damping is added to the system, whereas, if the settling time is prioritized, the system may become underdamped. Due to the optimization restriction ( $K_p > 0$ ), at no time did the system experience instability.

A frequency response analysis was performed with the control system tuned with the gain obtained from the proposed method ( $K_p = 0.1025$ ). The Figure 11 shows the Bode Diagram. The cutoff frequency is 1.59 Hz, the gain margin is infinite and the phase margin is



**Figure 10:** Step response for different values of  $\alpha$  and  $\beta$



74.67° (which proves the stability of the system). With this analysis is possible to conclude that the systems is not sensible to high frequency noise.

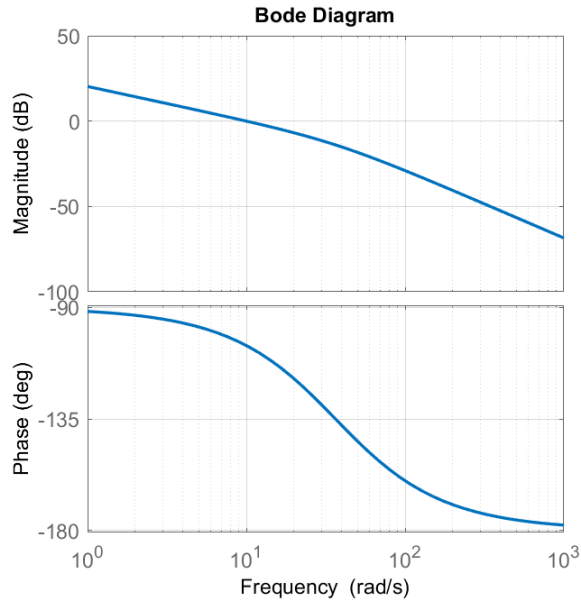


Figure 11: Bode diagram for the system with  $K_p = 0.1025$

A last test was performed with the system subject to a step input reference, a sudden disturbance type unit impulse and a sinusoidal noise with frequency of 200 Hz in the feedback branch. As the cutoff frequency of the system is 1.59 Hz all the noise was filtered. The position differences caused by the disturbance were eliminated, showing the of the control in tracking a reference and eliminate the effects of non-permanent disturbances. Both to follow the reference and to deal with disturbances, the system responded quickly, with a transient response of less than 0.5 s and null overshoot. Figure 12 depicts the system step response under noise and disturbance. The Equation (30) below describes the noise signal.

$$\delta(t) = 0.1\sin(1256.6t) \quad (30)$$

Considering the results obtained with this work, is possible to affirm that the hypothesis proposed in Section 1 is confirmed: with application of FSM, the  $K_p$  gain obtained made the system present a response with low settling time and negligible overshoot. The gain determination took place in less than a second (0.45 s), requiring only six iterations, which proves its low computational cost. The limitations of the Fibonacci Search

Methods are the need for determine the number of iterations previously and the fact of it find a interval that contains the optimizer point, and not the optimizer point properly. But its limitations are compensated by the ease of implementation of the algorithm and the use of center of mass to determine the optimizer.

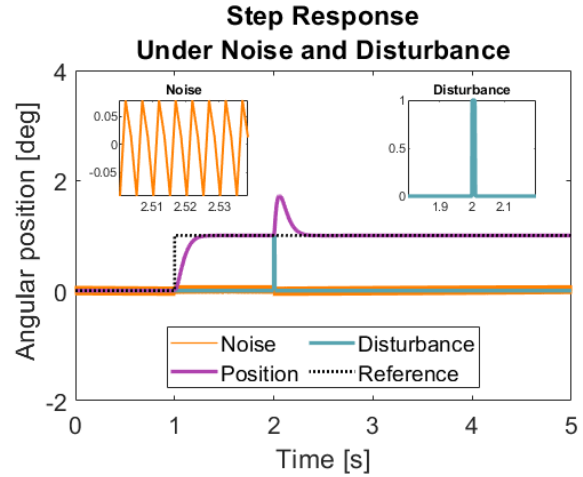


Figure 12: Step response of the system under noise and disturbance. The sinusoidal noise signal and the unit impulse disturbance were evidenced in the figure

## 8 Conclusions

In this work we proposed the use of Fibonacci Search Method (FSM) optimization algorithm to determine the  $K_p$  of a proportional controller applied to the position control of a DC motor. The project conditions were minimum settling time with null overshoot, that is, critical damping.

The results proved that the proposed method is able to determined the gain that satisfies the project requirements, with few iterations (in this work only six) in a short time (0.45 s). It was due the of the optimal contraction factor of the FSM, which results in a fast convergence.

## 9 Future Works

The uncertainties regarding the system parameters compromise the effectiveness of the optimization, since the objective function is related to such parameters. Thus, for future work, it is intended to develop a autotuning system in which the plant itself will be used, instead of its transfer function.

## References

- ACHANTA, R. K.; PAMULA, V. K. DC motor speed control using PID controller tuned by jaya optimization algorithm. In: *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*. Chennai, India: IEEE, 2017.
- CHONG, E. K. P.; ZAK, S. H. *An introduction to optimization*. 4. ed. New Jersey: John Wiley & Sons, 2012. 624 p. ISBN 978-1-118-27901-4.
- EVANS, W. R. Graphical analysis of control systems. *Transactions of the American Institute of Electrical Engineers*, Institute of Electrical and Electronics Engineers (IEEE), v. 67, n. 1, p. 547–551, jan 1948.
- FLORES-MORAN, E.; YANEZ-PAZMINO, W.; BARZOLA-MONTESES, J. Genetic algorithm and fuzzy self-tuning PID for DC motor position controllers. In: *2018 19th International Carpathian Control Conference (ICCC)*. Hungary: IEEE, 2018.
- GEE, W.; JAMES, F. A. J. L. *Sturgeon, William (1783–1850)*. Oxford University Press, 2004. Disponível em: <<https://www.oxforddnb.com/view/10.1093/ref:odnb/9780198614128.001.0001/odnb-9780198614128-e-26748>>.
- GHANY, M. A.; SHAMSELDIN, M. A.; GHANY, A. A. A novel fuzzy self tuning technique of single neuron PID controller for brushless DC motor. In: *2017 Nineteenth International Middle East Power Systems Conference (MEPCON)*. Nasr City, Egypt: IEEE, 2017.
- GVR. *Electric DC Motors Market Size, Share, Global Industry Report, 2025*. United States: Grand View Research, 2017. 160 p.
- HEKIMOGLU, B. Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm. *IEEE Access*, Institute of Electrical and Electronics Engineers (IEEE), v. 7, p. 38100–38114, 2019.
- HORLA, D.; SADALLA, T. Optimal tuning of fractional-order controllers based on fibonacci-search method. *ISA Transactions*, Elsevier BV, v. 104, p. 287–298, sep 2020.
- KOUASSI, B. A.; ZHANG, Y.; OUATTARA, S.; KIKI, M. J. M. PID tuning of chopper fed speed control of DC motor based on ant colony optimization algorithm. In: *2019 IEEE 3rd International Electrical and Energy Conference (CIEEC)*. Beijing, China: IEEE, 2019.
- MIRJALILI, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, Elsevier BV, v. 96, p. 120–133, mar 2016.
- MIRJALILI, S.; MIRJALILI, S. M.; LEWIS, A. Grey wolf optimizer. *Advances in Engineering Software*, Elsevier BV, v. 69, p. 46–61, mar 2014.
- RAMYA, M. A.; JADHAV, S. P.; PAWAR, S. N. Design and implementation of particle swarm optimization (PSO) tuned PID controller for speed control of permanent magnet brush less DC (PMBLDC) motor. In: *2020 International Conference for Emerging Technology (INCET)*. India: IEEE, 2020.
- RAO, S. S. *Engineering optimization: theory and practice*. 4. ed. New Jersey: John Wiley & Sons, 2009. 813 p. ISBN 978-0-470-18352-6.
- THAKUR, A.; MONGA, H.; SINGH, V. P.; KUMAR, R.; MATHUR, A. Sine cosine algorithm assisted tuning of PID controller for DC servo-motor. In: *2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM)*. Dubai, United Arab Emirates: IEEE, 2020.
- ZIEGLER, J. G.; NICHOLS, N. B. Optimum settings for automatic controllers. *Transactions of the ASME*, v. 64, p. 759–768, 1942.

## Appendix

DC motor parameters (based on the Quanser Rotary Servo Base Unit SRV02).

**Table 3:** DC motor parameters

Symbol	Value	Unit
$K_t$	0.00767	$N.m$
$K_m$	0.00767	$V/(deg/s)$
$R_m$	2.6	$\Omega$
$K_g$	70	
$B_{eq}$	$4e^{(-3)}$	$N.m.s$
$J_m$	$3.87e^{(-7)}$	$kg.m^2$
$J_{eq}$	$2e^{(-3)}$	$kg.m^2$
$\eta_m$	0.69	
$\eta_g$	0.9	

The quantities were defined in the text.