

## On the tuning of a PI speed controller for electric drives by using Simulated Annealing algorithm

Wander G. da Silva\*. Guilherme F. dos Santos\*. Volker Pickert\*\*

\* *School of Electrical, Mechanical and Computer Engineering – EMC  
Federal University of Goiás – UFG, Brazil*

\*\* *School of Engineering, Newcastle University, United Kingdom*

---

**Abstract:** This paper presents an investigation on the use of the Simulated Annealing (SA) algorithm to find the best tuning of a PI speed regulator for a speed control DC motor drive. Two control loops will be considered, an inner one for the armature current control and the outer one for the speed. The gains of the PI current regulator will be kept constant whereas the SA will be used to tune the speed regulator. The integral of the absolute of the speed error will be used as the evaluating function. Then, the faster the speed response reaches the speed reference for a load condition, the better the tuning. The range of the proportional and the integral gains will be limited such as the armature voltage and current does not exceed the rated value, keeping the system linear. Because of this, the best tuning of the speed controller can be easily predicted, and the SA algorithm will be put to the test. Two important SA parameters will be changed, what determines how fast the algorithm can converge towards the best solution. Simulation results will be presented, showing how accurate and fast SA can be to find the best tuning for the PI speed regulator.

**Keywords:** Speed control; electric drives; simulated annealing; DC motor.

---

### 1. INTRODUCTION

According to the Grand View Research company, the global electric DC motor market was worth US\$ 20 million in 2016. Their main application in industry ranges from handheld tools to the manufacture of strip-steel, paper or in the film making process, where accurate speed control is essential to ensure product quality. From the academic point of view, they are referred to in most of the Linear Control Theory books [Katsuhito Ogata, 2003; Norman Nise, 2000].

The closed loop speed control of a DC motor drive is usually realized by using the classical Proportional-Integral (PI) controller [W. G. da Silva, P. P. Acarnley and J. W. Finch, 2000, 2001]. However, every single variable speed DC motor drive depends on a variable DC voltage source which is obtained by using either, a Controlled Rectifier or Chopper. These power converters are naturally non-linear, although, due to their fast time response compared to the dynamics of the motor speed variation, can be linearized and regarded as a first order system with a time delay [R. Krishnan, 2001]. On the other hand, there is a couple of non-linearities which are not modelled, and one must have in mind the fact that the power converters output capability such as voltage and current, is limited. Because of such limitations, should the speed regulator, for a specific speed or torque demand, exceed the converter's capability, the classical control theory can no longer be used to find the tuning of the controller's gains. In such condition, different methods to adjust the controller's gains have already been proposed over the last 20 years [K. Sundareswaran and M. Vasu, 2000; W. G. da Silva, P. P. Acarnley and J. W. Finch, 2001 and 2004; Rohit G. Kanojiya

and P. M. Meshram, 2012; B. Mahesh Kumar and R. Babu Ashok, 2018; Tassneem Mohammed Reda, Karim Hassan Youssef, Ibrahim Fouad Elarabawy and Tamer Helmy Abdelhamid, 2018]. Furthermore, different control strategies have been proposed by many researchers worldwide. Among them is the use of fuzzy logic [W. G. da Silva, P. P. Acarnley and J. W. Finch, 2004; N. N. Baharudin and S. M. Ayob, 2015], artificial neural network [Tariq NN El-Balluq, Paul P. Acarnley and D. Atkinson, 2004], sliding-mode control [Shengxian Zhuang, Yulin He, and Senlin Wang, 2006], and Adaptive Backstepping Observer [A. Farrokh Payam and B. Mirzaeian Dehkordi, 2006; Yulin He and Senlin Wang, 2006]. However, every single control strategy has its own particularity and many times, the number of parameters to be optimised may become large. In the fuzzy speed control of electric drives, W. G. da Silva et al (2004) have used the minimum number of membership functions and rules for the fuzzy speed regulator of an electric drive. Even though, 16 parameters needed to be adjusted at the same time. Genetic Algorithm was used and, depending on the searching space, population size and generation number for instance, the optimization process could take long to reach to an end. Despite the strength of different optimization algorithm and, on the problem to be solved, some of the algorithm can fall into local minima.

Aiming to improve the efficiency of different optimisation algorithm for different problems, many researchers have put together the strengths of each one. Then, either a hybrid or improvements within the optimization algorithm have been proposed [Lin Xiuqin, Huang Jinfeng, Xu Yongpeng, Yan Ye, Zhang Yiming and Chen Xiaoxin, 2015; Haichen

Guo, Boyan Zhou, Pingping Yang and Xincheng Gu, 2017; Shigang Cui, Huimin Lv, Xingli Wu, Yongli Zhang and Lin He, 2018; Wang Yin-ling and Li Hua-cong, 2019].

In this paper, the PI speed controller for a DC motor drive will be tuned by using a Simulated Annealing (SA) algorithm. In another way, for comparison purpose, instead of using the classical control theory as suggested by Krishnan [R. Krishnan, 2001], the controller's proportional gain will be determined by using the knowledge of an expert such that, the rated values of armature voltage and current, for a specific speed demand and load torque, won't be exceeded. Furthermore, the controller's output will not be saturated at any time. By doing this, one can guarantee that the system will remain linear. The integral gain will then be chosen by the designer such as 10% speed response overshoot is accepted.

For the SA algorithm, the proportional gain will be limited to the value defined by the designer, ensuring that the rated armature voltage and current of the motor shall not be exceed. The range of the integral gain will be sufficiently large in order to turn the searching task for the SA algorithm a little more difficult.

## 2. MODELING OF THE DC MOTOR

The equivalent circuit of a separately excited DC motor is shown in Figure 1.

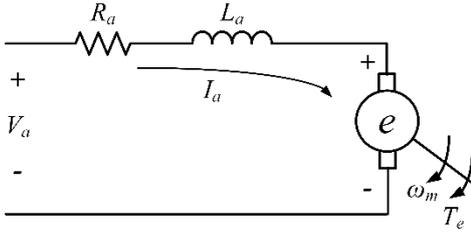


Fig. 1 Equivalent Circuit of the separately excited DC motor.

Where:

$V_a$  – Armature voltage [V]

$i_a$  – Armature current [A]

$R_a$  – Armature resistance [ $\Omega$ ]

$L_a$  – Armature inductance [H]

$e$  – Electromotive force – *fem* [V]

$\omega_m$  – Angular speed [rad/s]

$T_e$  – Electromagnetic torque [N·m]

By applying Kirchoff's voltage law to the circuit shown in Figure 1:

$$V_a = e + R_a i_a + L \frac{di_a}{dt} \quad (1)$$

The *fem* is a function of the field flux,  $\phi_f$ , the motor angular speed,  $\omega_m$ , and a constant,  $K$ , which is related to the machine design:

$$e = K \phi_f \omega_m \quad (2)$$

For the DC motor with constant field flux, (2) can be written as:

$$e = K_b \omega_m \quad (3)$$

Where  $K_b$  is the *fem* constant in V/rad/s.

The power balance equation states that:

$$V_a i_a = e i_a + R_a i_a^2 \quad (4)$$

From (4), one can clearly see that,  $V_a i_a$  represents the input power and,  $R_a i_a^2$ , the armature loss. Then, the remaining term,  $e \cdot i_a$ , is the electrical air gap power which will be transformed into mechanical one.

In terms of angular speed,  $\omega_m$ , and electromagnetic torque,  $T_e$ , the mechanical power can be given as:

$$P_a = \omega_m T_e = e i_a \quad (5)$$

By substituting (3) into (5), the electromagnetic torque can be found as:

$$T_e = K_b i_a \quad (6)$$

One can recognize that the *fem* constant,  $K_b$ , is the same one for the torque. However, in (8) it has the unity of N·m.

The load torque is usually modeled as a moment of inertia,  $J$ , and a viscous friction coefficient,  $B$ . Then, from the electromechanical modelling, the acceleration torque can be given as:

$$J \frac{d\omega_m}{dt} + B \omega_m = T_e - T_L = T_a \quad (7)$$

Where:

$J$  – Inertia [Kg.m<sup>2</sup>]

$B$  – Viscous friction coefficient [Nm/rad/s]

$T_L$  – Load torque [Nm]

$T_a$  – Acceleration torque [Nm]

By applying Laplace Transform to (1) and (7):

$$I_a(s) = \frac{V_a(s) - K_b \omega_m(s)}{L_a s + R_a} \quad (8)$$

And

$$\omega_m(s) = \frac{K_b I_a(s) - T_L(s)}{J s + B} \quad (9)$$

In terms of block diagram, (8) and (9) can be represented as in Figure 2.

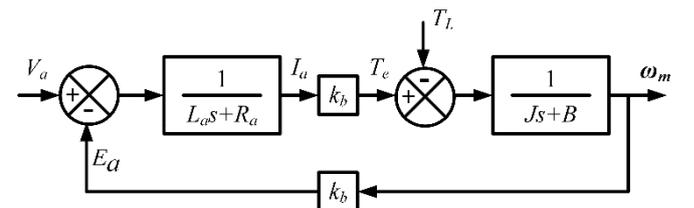


Fig. 2 Block diagram of the separately excited DC motor.

The two transfer, taking angular speed,  $\omega_m$ , as the output and armature voltage,  $V_a$ , and load torque,  $T_L$ , as the inputs, are:

$$\frac{\omega_m(s)}{V_a(s)} = \frac{K_b}{JL_a s^2 + (BL_a + JR_a)s + (BR_a + K_b^2)} = G_{\omega V}(s) \quad (10)$$

$$\frac{\omega_m(s)}{T_L(s)} = \frac{-(L_a s + R_a)}{JL_a s^2 + (BL_a + JR_a)s + (BR_a + K_b^2)} = G_{\omega L}(s) \quad (11)$$

The speed response, taking the two simultaneous inputs, armature voltage and load torque, is:

$$\omega_m(s) = G_{\omega V}(s)V_a(s) + G_{\omega L}(s)T_L(s) \quad (12)$$

The inverse Laplace transform of (12) yields to the time speed response of the DC motor due to the armature voltage and load torque, which is seen as a load disturbance. From (10) and (11), one can see that the separately excited DC motor represents a second order linear system.

### 3. THE CLOSED LOOP SPEED CONTROL

The speed control of the DC motor drive is usually carried out by using two control loops as shown in Figure 3. The inner one controls the armature current,  $I_a$ , and consequently, the electromagnetic torque,  $T_e$ . The outer loop controls the speed. The current control is done by comparing the actual armature current to a reference value,  $I_{ref}$ , generating a current error. The armature current error is then, taken into a PI current controller, which output is armature voltage to the motor. The speed control loop compares the actual motor speed to a reference value,  $\omega_{ref}$ . The speed error is then taken into a PI speed controller, which output is current demand to the current control loop.

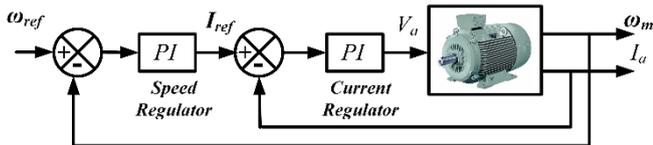


Fig. 3 Motor CC with closed loop speed control block diagram.

Where:

$\omega_{ref}$  – Reference speed.

$I_{ref}$  – Armature current demand.

As it can be seen from Figure 3, the PI current regulator output represents the armature voltage applied to the DC motor whereas the output of the speed regulator represents the armature current demand ( $I_{ref}$ ). It means that the armature voltage to be applied to the motor depends on the reference speed and load torque. Then, a source of variable DC voltage is needed to supply the DC motor, which can be either a AC-DC voltage converter for phase controlled DC motor drive [Krishnan, 2001], or a DC-DC pulse width modulated voltage converter (chopper) [Krishnan, 2001]. Both converters, from the control system point of view, can be modeled as linear first order system with a time delay [Krishnan, 2001]. However,

for the purpose of this paper, the power converter will be regarded as ideal with a unit constant gain. Then, the problem lies on finding the best tuning for both, the current and speed PI regulators.

One can find in the literature different approaches to the tuning of the PI regulators for a speed control DC motor drive. Krishnan (2001) gives a detailed guideline for both, to the tuning of the current and speed regulator. There is infinite possible gains adjustment which can guarantee control stability. In this paper, a practical way will be suggested, based on the reference speed and rated voltage and armature current of the DC motor.

The proportional gain of the current PI regulator will be the one such as, for a maximum value of current demand, which will be limited to the rated armature current, the proportional gain has to be the one that gives exactly the rated armature voltage. Then, the proportional gain of the PI current regulator must be equal to the rated voltage divided by the maximum current demand.

$$k_{pi} = \frac{V_{a(max)}}{I_{ref}} \quad (13)$$

Where:

$k_{pi}$  – current controller proportional gain.

$V_{a(max)}$  – maximum armature voltage.

The integral gain will be adjusted such as a current response overshoot of 20% will be accepted. It means that, from the control point of view, the current response settling time will be relatively faster than the underdamped or critically damped response.

The same reasoning will be applied to the tuning of the PI speed regulator. Since the output of the PI speed regulator cannot exceed the rated DC motor armature current, the proportional gain will depend on the maximum speed demand or reference speed,  $\omega_{ref}$ .

$$k_{p\omega} = \frac{I_{ref}}{\omega_{ref(max)}} \quad (14)$$

Where:

$k_{p\omega}$  – speed controller proportional gain.

$\omega_{ref(max)}$  – maximum reference speed.

Once again, the integral gain will be adjusted such as a speed response overshoot of 10% will be accepted. With such adjustment, the rated armature voltage and current will not be exceeded then, there is no need to add any non-linearity to the block diagram such as saturators at the output of the PI controllers. Consequently, there will be no integrator's *windup* [W. G. da Silva, P. P. Acarnley and J. W. Finch, 2001; Greeshma Sarah John and Abhilash T. Vijayan, 2017] and *anti-windup* circuits are not needed.

### 4. THE SIMULATED ANNEALING ALGORITHM

The Simulated Annealing algorithm was developed by Kirkpatrick [S. Kirkpatrick, C. C. Jr., Gelatt, and M. P. Vecchi, 1983] and, independently, by Cerny [Cerny, 1985]. It is defined as a stochastic search in which a potential solution to a problem is randomly generated,  $S'$ . This previously

generated solution is compared to an existing one,  $S$ . The probability of  $S'$  to be accepted is based on its proximity to  $S$ . Should  $S'$  be accepted, its suitability as a solution will then, be evaluated according to an exchange probability function and may be chosen to replace the previous one [S. A. Ethni, B. Zahawi, D. Giaouris and P. P. Acarnley, 2009]. The acceptance and exchange probability depend on a temperature parameter  $T$ . As the algorithm evolves,  $T$  is reduced according to a temperature coefficient,  $\alpha$ . Within the context of this paper,  $S'$  and  $S$  represent a set of proportional and integral gains of the PI speed regulator for the DC motor drive, which must be optimized by the SA:

$$S = (k_{p\omega}, k_{i\omega}) \quad (15)$$

$$S' = (k'_{p\omega}, k'_{i\omega}) \quad (16)$$

The possible values for both, the proportional and integral gains, are limited within the searching space.

The SA algorithm comes from the analogy to the physics where *annealing* can be understood as a process in which a solid is heated up to the point it becomes liquid (fusion point) and then, slowly cooled down until it becomes solid again. However, the cooling process must be sufficiently slow in order to keep the thermal balance in which the atoms will have enough time to reorganize themselves into a uniform structure [X. Yao, 1995]. The math basics to *annealing* comes from Boltzmann's distribution defined as:

$$P(i) = \frac{1}{N_0} e^{-\frac{E(i)}{kT}}, \forall i \in S \quad (17)$$

Where:

$P$  – Probability

$N_0$  – Normalize Constant

$E$  – Energy level

$k$  – Boltzmann Constant

$T$  – Temperature

$i = 1, \dots, n$

Initially, a sufficiently large thermo-dynamical system is considered. Within this context, the system can admit  $i$  possible states where each one has an associated energy level,  $E(i)$ . Then,  $P(i)$  represents the probability of the system to take an  $i$  energy state. Based on such theory, it is assumed that, when the atoms arrangement is stable, the probability of the system's energy to be  $E$  is proportional to  $e^{-E/kT}$ . Consequently, the probability of a system's energy to be  $(E+dE)$  can be found as:

$$\text{prob}(E + dE) = \text{prob}(E) e^{-\frac{dE}{kT}} \quad (18)$$

Where:

$$dE = E(i + 1) - E \quad (19)$$

It means that, as the temperature  $T$  decreases, according to the cooling coefficient  $\alpha$ , the probability of its energy state to

change is reduced to the point which represents the minimum energy state.

The SA algorithm can accept a new neighbouring solution even when it is worse than the current one. However, the acceptance probability becomes smaller as the larger is the distance between the new solution compared to the current one. It also applies to the temperature,  $T$ , i.e., the lower the temperature the smaller the probability of the new solution to be accepted when compared to the current one. These characteristics together guarantee that local minima are avoided.

Within this problem, the SA optimization algorithm will be used to find the best tuning of the PI speed regulator for the DC motor drive for which, the evaluation function will be based on the integral of the absolute of the speed error,  $I_{AE}$ . Then, the problem lies on minimizing  $I_{AE}$  by adjusting the proportional,  $k_{p\omega}$ , and integral,  $k_{i\omega}$ , gains of the PI speed regulator. The condition of accepting or rejecting a new solution, where  $I_{AE}$  would be increased, is determined by a sequence of random numbers with a limited probability given by  $e^{-\Delta/T}$ , where  $\Delta$  represents the difference between the two solutions, the current and the new one, and the control parameter,  $T$ , which represents temperature. It means that the larger the difference between the two solutions, the smaller the probability of the new one to be accepted.

The algorithm starts with high temperature value,  $T$ , and goes on testing several neighbouring possible solutions at each temperature level before it can be reduced.

Since the function to be minimised is the absolute of the speed error,  $I_{AE}$ ,  $\Delta$  is defined as:

$$\Delta = \frac{I'_{AE} - I_{AE}}{I_{AE}} \quad (20)$$

Depending on the initial values of the controller's gains, the  $I_{AE}$  can be relatively large. There is no suggestion in the literature of initial value for  $\Delta$ . However, in order to avoid an excessively large acceptance probability and keep the search limited to the neighbouring values,  $\Delta$  was chosen to be 4, which means 400% variation allowed within the neighbouring possible values of  $I_{AE}$ . Should  $\Delta \leq 0$ ,  $S'$  will be accepted as a new solution and will become the current one. Otherwise, a new solution will be accepted only if:

$$g(\Delta, T) > \text{random}(0,1) \quad (21)$$

Where:

$$g(\Delta, T) = e^{-\frac{\Delta}{T}} \quad (22)$$

It means that the SA algorithm will keep on searching for new possible solution while the temperature  $T$  is gradually reduced according to the temperature coefficient  $\alpha$ , until the probability to accept new solutions tends to zero and the system becomes stable. It was set up such as the current solution is constantly changed according to an acceptance probability. The new solution is generated at random within the limited searching space. It means that, for high values of  $T$ , the SA algorithm accepts a new solution which can be worse than the previous one, what is important to avoid local minima. On the other

hand, a global minimum can, eventually, be found within the early stages of the searching process and be replaced by a worse solution afterwards. In order to avoid losing the best solution,  $S^*$  was created to store the best result. It is expected that, at the end of the optimisation process, both,  $S$  and  $S^*$  shall be remarkably like each other but not necessarily the same. However, the probability of being different becomes smaller.

A pseudo SA algorithm is presented as follows:

```

S = (kpω, kiω)
Start
S* = S
iter = 0
T = T0
N = N0
  While T > Tolerance
    While iter < N
      iter = iter + 1
      Generate a neighbour S' within S neighbourhood
      Compute Δ = [f(S') - f(S)] / f(S)
      If Δ < 0 then S = S'
      If (f(S') < f(S*)), then S* = S'
      Otherwise, take x ∈ [0,1]
      If x < e-Δ/T then S = S'
    End
    T = T × α
    iter = 0
  End
Resume S*
End

```

## 5. PARAMETERS SET UP

In order to start the optimisation process by using the SA algorithm, some parameters must be defined. The DC motor parameters were obtained from Krishnan [R. Krishnan, 2001] as shown in Table 1.

Table 1. DC motor parameters

Symbol	Variable/parameter	Value
$V_a$	Armature Voltage	220 V
$R_a$	Armature Resistance	0,5 Ω
$L_a$	Armature inductance	3 mH
$J$	Inertia	0,0167 Kg·m <sup>2</sup>
$B$	Viscous coefficient	0,01 Nm/rad/s
$K_b$	Emf constant	0,8 V/rad/s
$K_t$	Torque constant	0,8 Nm/A

An armature current of 10A was taken as the rated value and a 5 Nm load torque as applied from the start.

Since the armature reference current,  $I_{ref}$ , was chosen to be equal to the rated armature current (10A), according to (13), the proportional gain of the PI current regulator must be 22 V/A. The integral gain was chosen to be equal to 100 V/A·s. These current controller gains will be kept unchanged.

The maximum reference speed was chosen to 100 rad/s. Then, according to (14), the maximum value for the proportional gain of the PI speed controller was 0,1 A/rad/s. The integral gain was chosen equal to 0,35 A/rad.

For the SA, the initial temperature,  $T_0$ , can be chosen in a couple of different ways, however, it must be sufficiently high, so the algorithm can explore a larger number of possible solutions. It is desirable the initial acceptance probability to be high, then,  $P(i)$ ,  $| i = 0$ , was chosen equal to 98% and  $\Delta_{max} = 400\%$ .  $T_0$  will then, be about 200K. The cooling factor,  $\alpha$ , is suggest such as  $0,8 < \alpha < 0,99$ . The closer to 1 the slower is the cooling down and the execution time of the SA algorithm. The smaller it is, the faster the searching process reaches to an end. On the other hand, the bigger is the chance the result does not represent to the best solution. Within this context, two different values will be tested,  $\alpha = 0,6$  and  $\alpha = 0,95$ .

The amount of tested solutions within each temperature level is determined by the parameter  $N$ . This parameter is highly dependent on the problem to which the best solution is to be found. Two different possible values will be tested,  $N = 1$  and  $N = 4$ .

The tolerance defines how low the final temperature  $T$  can be. Since the function  $e^{-1/x}$  decreases very quickly for small values of  $x$ , it does not need to be too low. Then, a tolerance of 0,01% will be chosen. The speed control DC motor block diagram as well as the SA algorithm was developed within the MATLAB/SIMULINK® environment. The DC motor model simulation time window was set to 4s, which was enough for the motor to speed up and settle at steady state.

## 6. RESULTS

The DC motor block diagram as shown in Figure 3 was run with the speed controller gains adjusted by the designer. As it can be seen, the speed response was underdamped with 10% speed overshoot. The armature voltage,  $V_a$ , at  $t=0s$  was exactly 220V (rated value) whereas the armature current,  $I_a$ , equal to 10A, as expected. Naturally, because of the overshoot, the armature current goes above the rated value (10A) for a short period of time before settling to the final steady state value. The final value of the absolute of the integral of the speed error,  $I_{AE}$ , was 35.3789 rad·s.

In order to investigate the SA algorithm capability to find the best tuning for the PI speed controller, it was run with different parameter setting. Depending on the settings, the iteration number necessary to the algorithm to converge to an optimum solution can become relatively large. On the other hand, the larger the iterations number, the larger the probability to converge to an even better result.

For the tuning of the PI speed controller, only two parameters must be adjusted, the proportional and the integral gains. Within this problem,  $T_0$  will be set to 200K and  $\Delta < 400\%$ . The DC motor drive Simulink model will be run within the same 4s time window. Two SA algorithm parameters have major impact on the convergence time of the SA algorithm,  $N$  and  $\alpha$ . The first run will be done with  $N = 1$  and  $\alpha = 0,6$ . The proportional gain range was set to  $0 < K_{p\omega} < 0,1$  A/rad/s whereas the integral one was set to  $0 < K_{i\omega} < 1$  A/rad. The results are as follows.

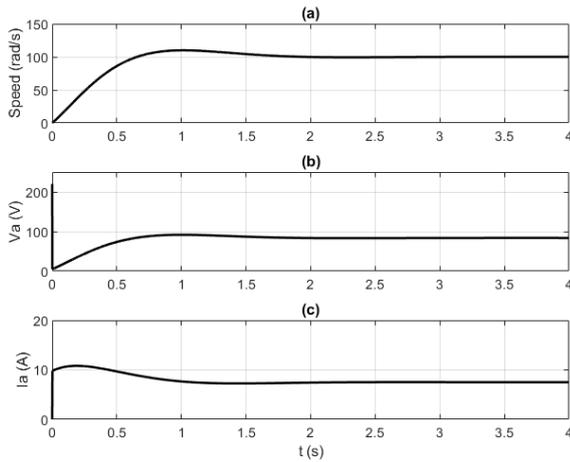


Fig. 4 DC motor speed (a), armature voltage (b) and current (c) responses with the speed controller's gains adjusted by the designer, for 100 rad/s step input reference speed.

Figure 5(a) shows how the speed controller gains evolved from the beginning to the end of the SA algorithm whereas Figure 5(b) depicts the development of the integral of the absolute speed error,  $I_{AE}$ , which ended after 19 iterations. As it can be seen, there was no evolution after the 13<sup>th</sup> iteration, what means that there was no improvement on the final result. From the control point of view, it was expected that the fastest response would demand the highest possible proportional gain which imposes the maximum armature current demand. The integral gain should be high enough to bring the speed to its final steady state value within the shortest of time. However, an excessively large integral gain would make the system oscillatory and having a long settling time. It means that, despite the small number of iterations, SA was capable of find a nearly optimal setting for the PI speed controller.

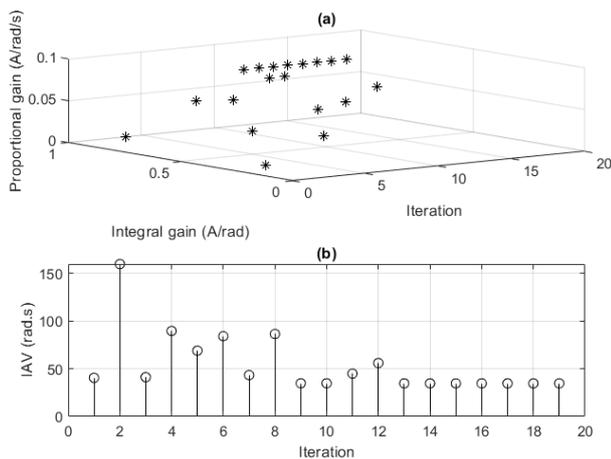


Fig. 5 PI speed controller's gains (a) and integral of the absolute speed error (b) given by the SA algorithm with  $N = 1$  and  $\alpha = 0,6$ .

Table 2 displays the best (global) results and the final one given by the SA algorithm. The integral of the absolute speed error with the setting given by the designer was  $35.3789 \text{ rad.s}$ , while the one given by the SA was  $34.5700 \text{ rad.s}$ . It means that the global solution found by the SA after 19 generation was

slightly better than the one given by the designer. With such setting, the SA algorithm converges relatively quickly. However, as it can be seen from the values of thee evaluation function,  $I_{AV}$ , its value is still changing between the iterations, what means that it is likely the that the best solution was not yet found. Nevertheless, it is also important to realise that, since the SA algorithm keeps on searching the neighbourhood around the best solution, the final result may not exactly match the global one. Furthermore, because of the stochastic search optimisation method, at each time the algorithm is run, especially for a small iteration number, the final result can be slightly different from one given by a previous run. By looking at Figure 5(a), one can see that the location of the proportional and integral gains was converging towards their limit within the searching space, which was an indication where the best results would lie.

Table 2

Comparison for  $N=1$  and  $\alpha=0,6$

Global Solution	Final Solution
$K_{p\omega}^* = 0,0661$	$K_{p\omega} = 0,0661$
$K_{i\omega}^* = 0,9986$	$K_{i\omega} = 0,9986$
$I_{AE}^* = 34,5700$	$I_{AE} = 34,5700$

Figure 6 depicts the best speed response (a) given by the best controller's tuning found by the SA algorithm with  $N=1$  and  $\alpha=0,6$ , together with the correspondent armature voltage (b) and current (c).

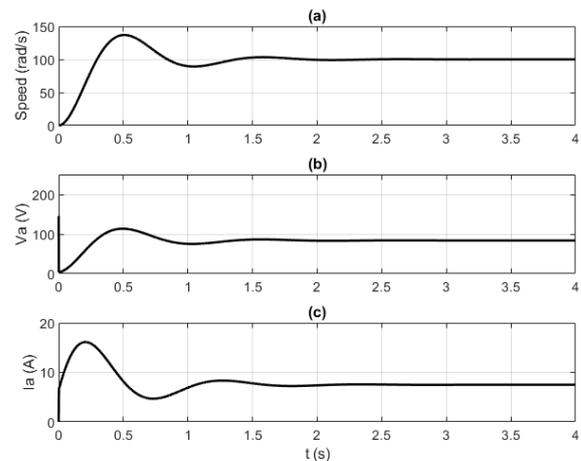


Fig. 6 Speed controlled DC motor speed (a) and armature voltage (b) and current (c) responses with the speed controller's gains adjusted by SA with  $N = 1$  and  $\alpha = 0,6$ .

Next will be shown another run with a lot larger iteration number, which results will be compared to those obtained in the previous run.

Figure 7 depicts the best speed response given by the best controller's tuning found by the SA algorithm with  $N=4$  and  $\alpha=0,95$ ). It can be clearly seen the best setting is around the region of the highest proportional gain, which guarantees faster speed response. The integral gain converged towards one value high enough to speed up the motor with the fastest settling time.

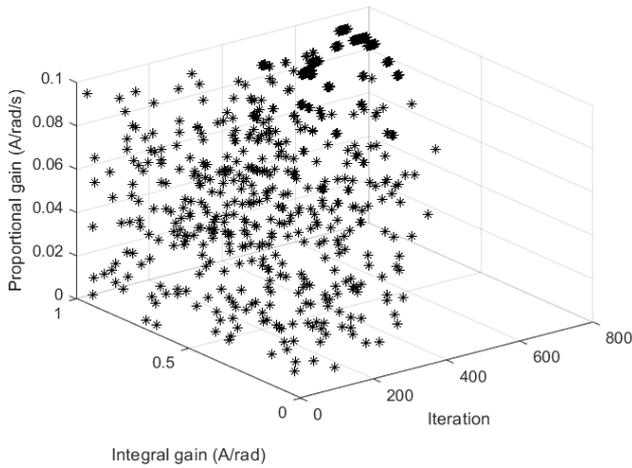


Fig. 7 PI speed controller's gains given by the SA algorithm with  $N = 4$  and  $\alpha = 0,95$ .

Figure 8 (a) illustrates the evolution of the integral of the absolute speed error,  $I_{AE}$ , while (b), in enlarged scale, gives a detailed view of the SA evolution after the 680<sup>th</sup> iteration. As it can be seen, the progress after that point was relatively small, with a little improvement after the 711<sup>th</sup> iteration, from where there were no further progress.

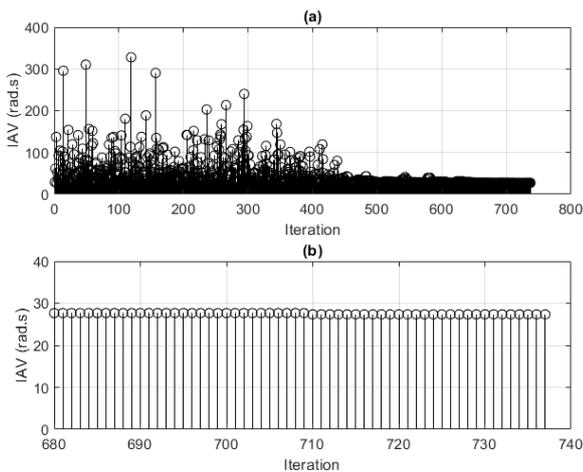


Fig. 8 Integral of the absolute speed error gains given by the SA algorithm with  $N = 4$  and  $\alpha = 0,95$ .

Table 3 displays the best (global) results and the final one given by the SA algorithm after 737 iterations. The integral of the absolute speed error ( $I_{AE}=26,8634 \text{ rad}\cdot\text{s}$ ), with such setting, was significantly smaller than the one given by the designer ( $I_{AE}= 35.3789 \text{ rad}\cdot\text{s}$ ) and the other obtained within the previous one ( $I_{AE}=34.5700 \text{ rad}\cdot\text{s}$ ).

Table 3  
Comparison for  $N=4$  and  $\alpha=0,95$

Global Solution	Final Solution
$K_{p\omega}^* = 0,0965$	$K_{p\omega} = 0,0965$
$K_{i\omega}^* = 0,9575$	$K_{i\omega} = 0,9575$
$I_{AE}^* = 26,8434$	$I_{AE} = 26,8434$

Figure 9 illustrates the behaviour the speed response (a), together with the armature voltage (b) and current (c).

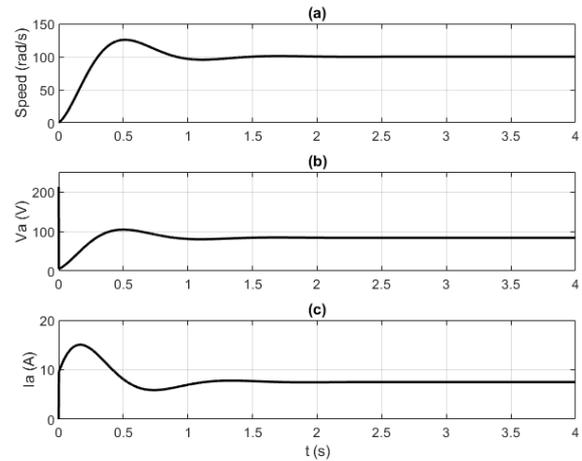


Fig. 9 Speed controlled DC motor speed (a) and armature voltage (b) and current (c) responses with the speed controller's gains adjusted by SA with  $N = 4$  and  $\alpha = 0,95$ .

Figure 10, the three speed responses together, showing the dynamics of each one. After analysing the results, one can realise that, by using the integral of the absolute speed error,  $I_{AE}$ , as the parameter to be followed to choose the best tuning, the proportional gain has to be as high as possible since it guarantees the fastest response by imposing a higher armature current demand. The integral gain must be high enough to ensure faster settling time, despite a possible overshoot.

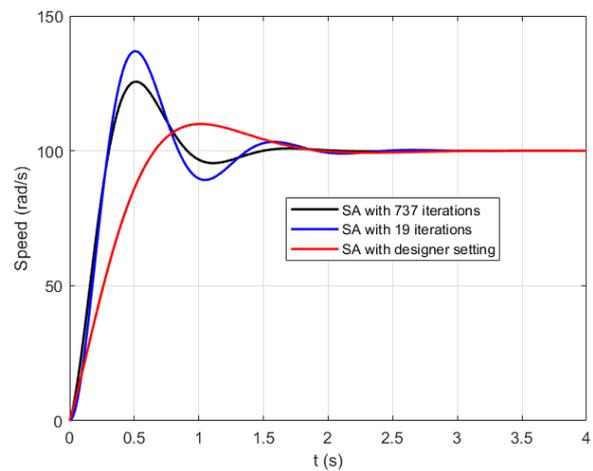


Fig. 10 Speed responses given by the designer, SA after 19 iterations ( $N=1$ ;  $\alpha=0,6$ ) e SA after 737 iterations ( $N=4$ ;  $\alpha=0,95$ ).

## 7. CONCLUSION

This paper has investigated the capability of the SA Algorithm to find the best tuning for a PI speed regulator for a speed controlled DC motor drive. The classical linear model of the DC motor was developed and two control loops were used, an inner one for the armature current and the outer one for the speed. The gains of the PI current control loop were kept constant and the SA was used to tune the gains the PI speed

regulator only. The possible gains values, which represents the searching space for the SA, were limited such that the system would remain linear. The criteria used to evaluate the best tuning was the integral of the absolute of the speed error. Then, the best tuning should be the one which makes the motor speed up and reach steady state within the shortest time. The best tuning given by the SA algorithm was such that the proportional and integral gain were both high, nearly hitting their limits, regardless a little speed overshoot. From the control point of view, it was expected since the higher the proportional gain the higher the initial armature current demand and, consequently, the acceleration torque of the DC motor. The higher the integral gain, the larger the speed overshoot. The lower the integral gain the longer the settling time. Regarding to the SA algorithm parameters, only the number of tests within each temperature level and the temperature coefficient were changed. The smaller the number of tests and the coefficient temperature, the faster the SA converges towards the final solution. However, the smaller the guarantee that the final solution would be the best possible one.

## REFERENCES

- A. Farrokh Payam, and B. Mirzaeian Dehkordi (2006). Nonlinear Sliding-Mode Controller for Sensorless Speed Control of DC servo Motor Using Adaptive Backstepping Observer. International Conference on Power Electronic, Drives and Energy Systems.
- B. Mahesh Kumar, and R. Babu Ashok (2018). Soft Computing Using GWO (Grey Wolf Optimization) for the Performance Improvement of High Speed Brushless DC Motor. International Conference on Emerging Trends and Innovations in Engineering and Technological Research (ICETIETR).
- Haichen Guo, Boyan Zhou, Pingping Yang and Xincheng Gu, (2017). Application of modified Stribeck model and simulated annealing genetic algorithm in friction parameter identification. 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE).
- K. Sundareswaran, and M. Vasu (2000). Genetic tuning of PI controller for speed control of DC motor drive. Proceedings of IEEE International Conference on Industrial Technology. Volume 1.
- Katsuhito Ogata (2003). *Engenharia de Controle Moderno*, 4<sup>a</sup> Edição, Pearson Education, São Paulo.
- Lin Xiuqin, Huang Jinfeng, Xu Yongpeng, Yan Ye, Zhang Yiming, and Chen Xiaoxin (2015). Clustering Analysis of Acoustic Emission Signals in 2D-C/SiC Tensile Damage Using Genetic Simulated Annealing Optimization Algorithm. 2015 IEEE 5th International Conference on Electronics Information and Emergency Communication.
- N. N. Baharudin, and S. M. Ayob (2015). Brushless DC motor drive control using Single Input Fuzzy PI Controller (SIFPIC). Conference on Energy Conversion (CENCON).
- Norman Nise (2000). *Control Systems Engineering*, 4<sup>th</sup> Edition, John Wiley & Sons, New York.
- R. Krishnan (2001). *Electric Motor Drives – Modeling, Analysis, and Control*, Prentice-Hall.
- Rohit G. Kanojiya, and P. M. Meshram (2012). Optimal tuning of PI controller for speed control of DC motor drive using particle swarm optimization. International Conference on Advances in Power Conversion and Energy Technologies (APCET).
- S. A. Ethni, B. Zahawi, D. Giaouris and P. P. Acarnley (2009). Comparison of Particle Swarm and Simulated Annealing Algorithms for Induction Motor Fault Identification. 7th IEEE International Conference on Industrial Informatics (INDIN 2009).
- S. Kirkpatrick, C. D. Jr. GELATT, and M. P. Vecchi (1983). Optimization by Simulated annealing. *Science*, n. 220, 1983, p. 671-680.
- Shengxian Zhuang, Yulin He, and Senlin Wang (2006). Fuzzy Sliding-Mode Control with Torque Observer in Induction Motor Drive. 6th World Congress on Intelligent Control and Automation. Volume 2.
- Shigang Cui, Huimin Lv, Xingli Wu, Yongli Zhang, and Lin He (2018). Optimization of plant light source based on simulated annealing particle swarm optimization algorithm. 2018 Chinese Control And Decision Conference (CCDC).
- Tariq NN El-Balluq, Paul P. Acarnley and David J. Atkinson (2004). Controller for a Brushless DC Motor: Optimum Speed Control using a Neural Network. IEEE International Symposium on Industrial Electronics.
- Tassneem Mohammed Reda, Karim Hassan Youssef, Ibrahim Fouad Elarabawy, and Tamer Helmy Abdelhamid (2018). Comparison Between Optimization of PI Parameters for Speed Controller of PMSM by Using Particle Swarm and Cuttlefish Optimization. Twentieth International Middle East Power Systems Conference (MEPCON).
- V. Cerny (1985). Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *Journal of Optimisation Theory and Application*, n. 45, 1985, p. 41-51.
- W. G. da Silva, P. P. Acarnley, and J. W. Finch (2000). Application of genetic algorithms to the online tuning of electric drive speed controllers. *IEEE Transactions on Industrial Electronics*, Volume 47, Issue 1.
- W. G. da Silva, P. P. Acarnley, and J. W. Finch (2001). Anti-windup Circuits with On-line Optimisation by Genetic Algorithm. 9th European Conference on Power Electronics and Applications – EPE 2001, Graz, Austria.
- W. G. da Silva, P. P. Acarnley, and J. W. Finch (2004). On-line optimisation of a fuzzy drive controller using genetic algorithm. 2004 IEEE International Symposium on Industrial Electronics, Volume 2.
- X. Yao (1995). A new simulated annealing algorithm. *International Journal of Computer Mathematics*, 56, p. 161-168.