

WIND TURBINE CONTROLLER TUNING BY GENETIC ALGORITHM

Decio Crisol Donha
University of São Paulo-Mechanical Eng. Depart.-Av. Prof. Mello Moraes-2231 (05508-900) São Paulo-Brazil.
decdonha@usp.br

Gerson Risso
University of São Paulo-Mechanical Eng. Depart.-Av. Prof. Mello Moraes-2231 (05508-900) São Paulo-Brazil.
grisso@usp.br

Abstract. *A tuning procedure for classical controllers by means of Genetic Algorithms is considered. The method is applied to the synthesis of an optimal PID controller of a wind turbine operating under different weather conditions. Results of simulations are presented, analyzed and compared to previous works.*

Keywords: *Genetic algorithm, Wind Turbine, PID, Multi-objective optimisation, Controller tuning.*

1. Introduction

The electrical energy crisis of 2001 showed that Brazil will have very soon to look for new sources of production. Due to environmental reasons, these sources must no only be cost competitive, but renewable and clean. In this context, eolic energy appears to be one of the most interesting options, due to the large wind fields in the country, which remain virtually unexplored. Considering also that in these fields the hydraulic is complementary to the eolic climate cycle, this option is certainly one of the most promising technical and economical solution at hand.

Eolic energy is nowadays usually recovered by Horizontal Axis Wind Turbines (HAWT) or by Vertical Axis Wind Turbines (VAWT), each type showing very distinctive characteristics. Although studied for a long time in other countries, the problem associated with wind turbine control systems remains undisclosed and is at once interesting and very challenging, as shows the extensive bibliography about the issue [De La Salle,1990].

This work is concerned with the tuning of a classical *PID* controller used to optimize the electrical energy production of a 300 KW HAWT [Anderson,1991]. In the literature one finds a number of tuning methods for classical controllers, such as, among others, the Ziegler-Nichols methods, the tuning by optimization based on performance indices (ISE, IAE, ITAE, ITSE) or pole placement procedures [Dorf,1993]. All these methods show some drawback, and are usually employed to a limited class of problems [Aström,1988]. In this work, we propose an alternative automatic tuning method for a *PID* controller by means of a Genetic Algorithm (*GA*). This algorithm is used to generate a huge number of possible solutions for the three gains of the controller, which is then evaluated and selected by the optimization of a performance index. Due to proper features of *GA*s, such as to concentrate in the most promising solutions, the search is quite easy and fast, and in general produces very satisfactory results [Goldberg,1989], [Mitchel,1999].

This paper is organized as follows: in section two the control problem of a HAWT is posed and a control loop is presented; section three presents a quick summary of *GA* features and operators; in section four, results of two case studies are presented, analyzed and compared to results obtained by other authors; in section five conclusions are drawn and suggestions for future researches are presented.

2. Wind Turbine Control Problem

In a HAWT, the produced power is proportional to the cube of the wind speed. Since wind speeds over $30,0\ m/s$ are rare, and energy recovering over this limit would lead to the necessity of high performance drive trains and structures, with clear cost implications, it is usual to employ aerodynamic power limiters [Novac,1995] and [Spera,1994]. These devices prevent overload and allow the operation of the turbine until a maximum speed, when the turbine is switched-out. A typical operation curve is shown in Fig. 1. Not considering a passive regulation through the stoll of the turbine blades, it is usual to achieve an active regulation through the control of the pitch angle of the turbine blades [Anderson,1990], [Anderson,1991], [Grimble,1992], [Grimble,1991].

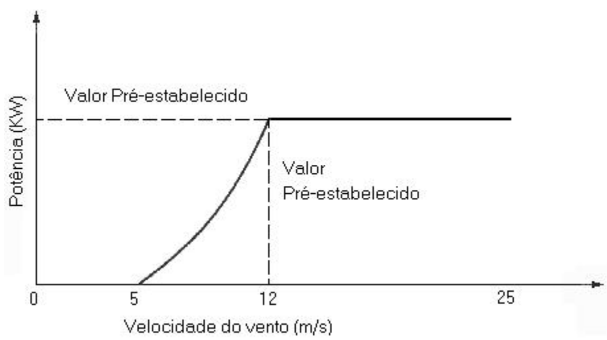

Figure 1: Wind Turbine Power Limit

The system under consideration may be well modelled by a scheme similar to that shown in Fig. 2.

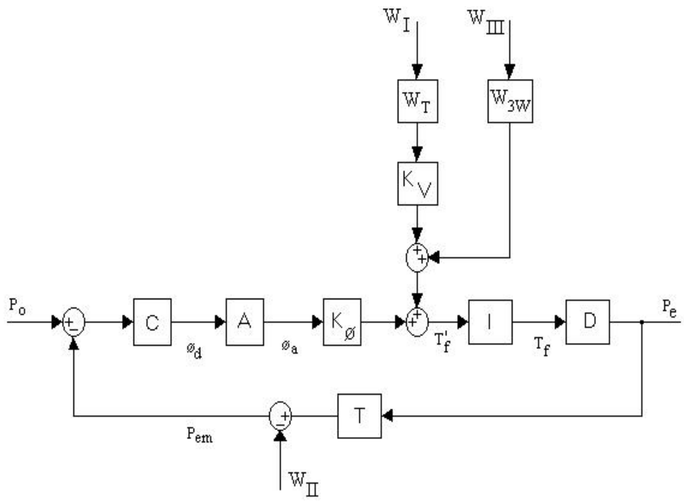

Figure 2: Control System Diagram

The scheme presented in Fig. 2 was obtained from former modelling by experienced researchers [Anderson,1990]. Although linear, it includes the most significant involved phenomena such as the turbulent wind disturbances, the rotational disturbance induced by interaction wind-structure, sensor noises and induction lag (I), the time lag of the system for a new aerodynamic condition. Notice that the system is intended to maintain the actual power level P_e as close as possible to the required power P_o and that the wind force enters the system as a disturbance. The actuator (A), the induction lag and the drive train (D) are transfer functions related to a three blade, 300 KW, constant speed HAWT. Following well-known guidelines to achieve a simple but well representative model [Anderson,1990] and [Anderson,1991], disturbances and noises are generated by a convenient filtering of white noises (ω_I, ω_{II} and ω_{III}) by convenient transfer functions (W_T - turbulent wind, $W_{3\omega}$ - rotational structure disturbance). In Fig. 2, C is a three term *PID* controller, which gains are to be defined, T is a transducer working as a sensor and K_v and K_ϕ are speed variable gains obtained from a look-up table. Transfer functions of each loop element are presented in [De La Salle,1990] and [Grimble,1992].

3. Genetic Algorithm

Genetic Algorithms first appeared with the pioneering work of John Holland and collaborators [Holland,1992a] and [Holland,1992b], who were emulating adaption in nature on computer programs. The research results brought significant innovations relatively to other evolutionary computer methods and immediately caught great attention and was promptly extended to more general problems [Goldberg,1989].

A simple *GA* is usually started with a population of individuals (actually binary or decimal strings), which evolves to new and better populations (according to a performance index) by means of operators also inspired by nature, such as reproduction, crossover, mutation and inversion [Cao,1999], [Goldberg,1989] and [Mitchel,1999]. The procedure can be stopped after a pre-defined number of generations or before that if a very good solution is found.

Using a Darwinian survival-of-the-fittest analogy *GA*s can eliminate unfit population member characteristics and lead to fast convergence to optimal solutions. This involves searching high-dimensional spaces for superior, if not optimal solutions. The computer intensive algorithms used are simple, robust and general. No knowledge of the search space, other than its limits, is usually assumed. *GA*s are parallel, global search procedures allowing for the simultaneous evaluation of many points in the universe of solutions which are more likely to converge toward the global solution. Assumptions such as search space differentiability and continuity or knowledge of parameters and problem structure are no longer an issue.

3.1. Reproduction

The reproduction process can be subdivided into two subprocesses: Fitness Evaluation and Selection. The fitness function is what drives the evolutionary process and its purpose is to determine how well a string (individual) solves the problem, allowing for the assessment of the relative performance of each population member. In this work the fitness evaluation was established by a suitable weighted function, which takes in account system response parameters (rising time, settling time, offset and overshoot) to a step input as follows:

$$\min \quad \mu_A(X) = \sum_{i=1}^{k} \alpha_i \mu_{fi}(X) \tag{1}$$

where $X \in R^n$ is any possible solution; $\mu_\bullet$ is the fitness value of a parameter; α_i are weights attributed to a parameter $f_i(X)$. This means that an optimal solution is found to satisfy an a priori defined fitness function.

The selection operator chooses individuals of the population, based on the fitness evaluation, that will later generate offsprings of future populations. In this work the Tournament Selection with an Elitist Strategy [KrishnaKumar,1994] was employed to implement the selection operator.

If an entire population has been evaluated and a suitable solution was not found, a new generation is created.

Reproduction cannot create new and better strings. This improvement may be achieved by Crossover and to a lesser extent by Mutation.

3.2. Crossover

Reproduction may proceed in three steps as follows: a) two newly reproduced strings are randomly selected from a Mating Pool; b) a number of crossover positions along each string are uniformly selected at random and c) two new strings are created and copied to the next generation by swapping string characters between the crossover positions defined before.

3.3. Mutation

The mutation operator is a secondary mechanism of *GA* adaption and is only introduced to provide a framework to ensure that a critical feature (genetic information) may be reinserted or removed from a population. Mutation generates new individuals by randomly modification of the value of a string position (gene).

4. Case Studies

Several simulations were conducted to validate the proposed method. Before applying the optimization method to the turbine control problem, the performance of *PID* controllers tuned by *GA*s were compared to the similar controllers tuned by classical methods such as the Ziegler and Nichols and *ITAE*, using simple plants.

In most of the simulations, the performance index was based on the minimization of the system response parameters to a step input, which well simulates a sudden change in operation conditions. A typical performance index is given by:

$$J = \alpha t_r + \beta t_s + \gamma M_p + \eta e_{ss}$$

where α, β, γ and η are weighting factors, imposed by the user to achieve desired response characteristics; t_r is rise time; t_s is the settling time; M_p is the overshoot and e_{ss} is the steady-state error.

To illustrate the validation process, a *PID* controller for a second order plant, is tuned using the alternative methods. The velocity control of an eletric motor is the problem at hands. The plant transfer function is given by [Dorf,1993]:

$$G_p(s) = \frac{1}{s^2 + 2s + 4}$$

The control system was then tuned by *GA* and by as ITAE controller, using a prefilter to reduce overshoot. Figure 3 presents a typical system response to a step input using the using the controller tuned by the *GA*, wheras Fig. 4 presents the response with the ITAE controller. GA parameterrs where set as follows: 20 Generations; 21 individuals per Generation; 105 bits per chromossome; 0.001 of mutation probability; 0.95 of crossover probability; two crossing points per chromossome; tournament selection; $\alpha = 5$; $\beta = 10$; $\gamma = 0.9$; $\eta = 10$. Table 1 shows that similar results were obtained is this case.

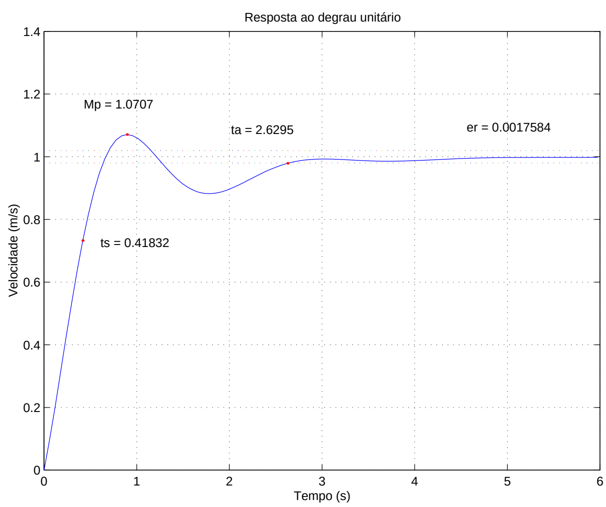

Figure 3: Validation: Servo tuned by GA

Table 1: Comparison ITAE and GA

Method	$t_r(s)$	$t_s(s)$	M_p	$e_{ss}(\%)$
ITAE	0.8	2.1	0.02	0.0
AG	0.4	2.6	0.07	0.2

A more comprehensive validation and comparison may be found in [Risso, 2002].

Next, considering the wind turbine control problem, two case studies are presented and comparisons between *GA* tuned controllers and classical controllers are carried out. In both cases, the main control objective is to keep the produced power near a desired value, despite changes in the wind speed. In both cases the following *GA* parameters were used: 30 Generations; 21 individuals by Generation; 126 bits as size for each individual (chromosome); 0.001 for the mutation probability; 0.97 for the crossover probability and two points for crossover. For selection purposes, the wheel of fortune with a sigma truncation method to avoid a premature convergence ([Mitchel,1999]), and alternatively a tournament selection with elitism were used.

Typical values of weighting factors are now $\alpha = \beta = 1$, $\gamma = 50$ and $\eta = 200$, indicating that the overshoot and the steady state error are considered the most important characteristics.

4.1. Case Study One

In this case, the average nominal wind speed (15.7 m/s), for 12° pitch-angle is used. The *PID* controller transfer function picked by the *GA* is as follows:

$$G_c(s) = \frac{2.1s^2 + 2.2s + 5.3}{10^3 s}$$

The step response is given in Fig. 5, where one notices a quick response ($t_r = 0.11s$ and $t_s = 3.13s$) with zero overshoot and a very small steady state error ($e_{ss} = 0.55\%$). Figure 6, shows that the convergence occurs in the 3^{rd} generation, with a performance index $J = 54.4$. Figure 7 shows another run for the same conditions, when the following *PID* controller was found:

$$G_c(s) = \frac{0.1s^2 + 2.6s + 4.5}{10^3 s}$$

A similar response is obtained with this controller ($t_r = 0.13s$ and $t_s = 3.14s$), and no considerable overshoot or steady state error are observed again. Figure 8 shows that the *GA* converges around the 70^{th} generation, with $J \approx 54.2$.

Figure 9 shows the result obtained by Grimble [Grimble,1992] using a *PI* controller. The *D* term was dropped out due to tuning difficulties, and therefore slow responses are expected in this case. Indeed, Fig. 9 shows a slower response ($t_r = 6.9s$ and $t_s = 31.5s$), also is this case with zero overshoot and steady state error. Table 2 is intended to easy the comparison, still showing a different GA search result. No doubt that the procedure leaded to a clear optimization.

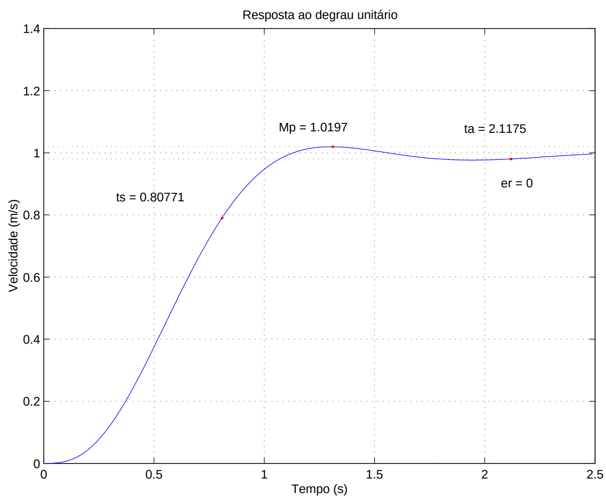

Figure 4: Validation: Servo tuned by ITAE

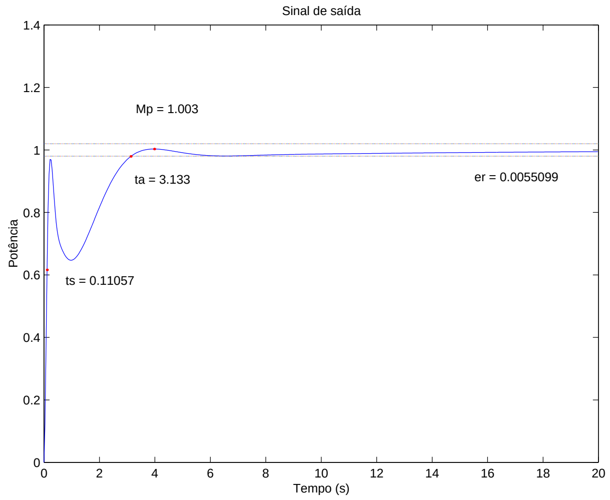

Figure 5: Case 1: step response GA run 1

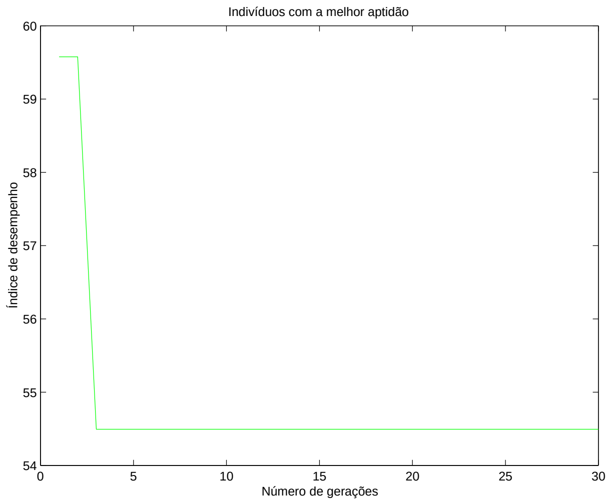

Figure 6: Case 1: GA convergence run 1

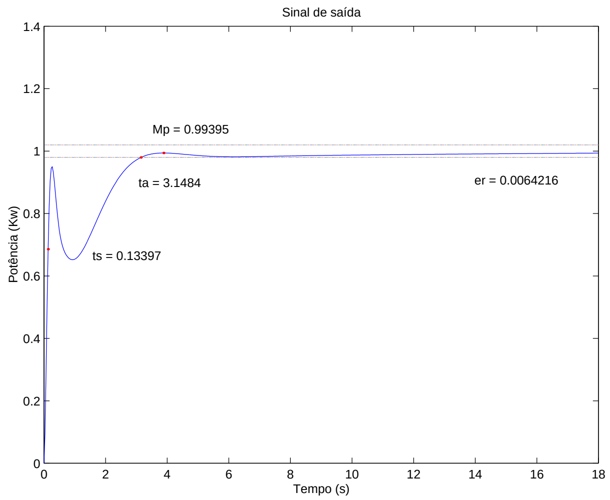

Figure 7: Case 1: step response GA run 3

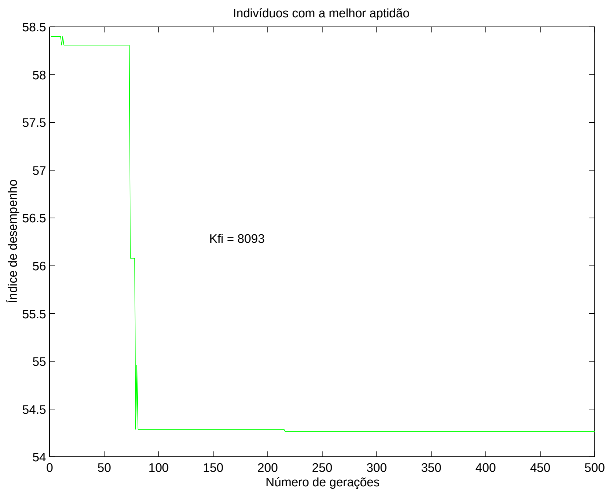

Figure 8: Case 1: GA convergence run 3

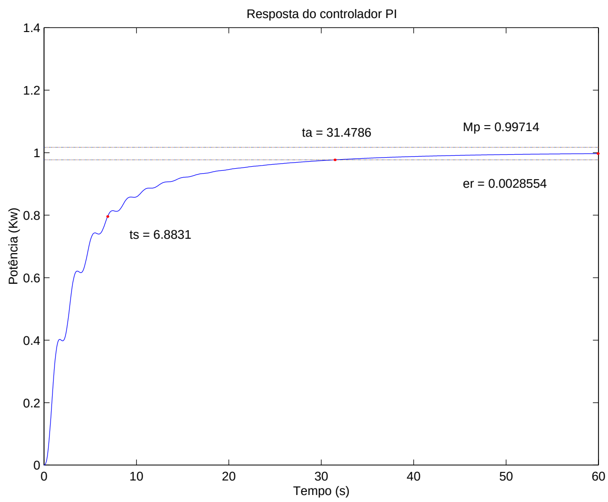

Figure 9: Case 1: PI Controller by Grimble et al.

Table 2: Case 1 comparison

Controller	*PI*	*GA 1*	*GA 2*
$t_r(s)$	6.9	0.2	0.4
$t_s(s)$	31.5	2.9	7.0
M_p	0.0	0.0	0.0
$e_{ss}(\%)$	0.002	0.008	0.006
J	–	54.4	58.3

4.2. Case Study Two

To show the strength of the method, the operation out of nominal conditions is investigated in this case. The wind speed is now 12.6 m/s, which correspondes to a 4° blade pitch angle.

Figure 10 is produced with the following controller tuned by the *GA* procedure:

$$G_c(s) = \frac{4.61s^2 + 0.4s + 4.5}{10^3 s}$$

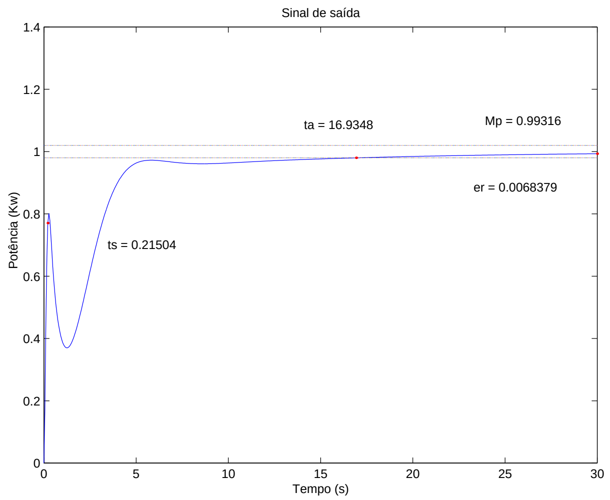

Figure 10: Case 2: step response GA run 1

Figure 11 shows the algorithm convergence occurring at the 4^{th} generation with $J = 68.2$.

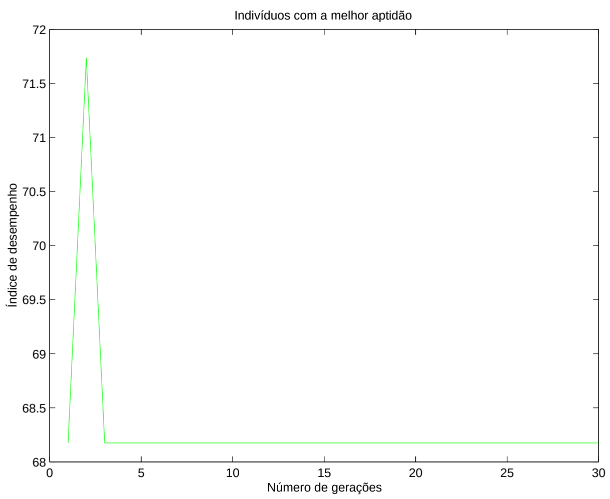

Figure 11: Case 2: GA convergence run 1

Another run of the algorithm leaded to the following *PID* controller:

$$G_c(s) = \frac{4.9s^2 + 0.8s + 5.3}{10^3 s}$$

Results obtained is this case are presented in Fig. 12 and in Fig. 13.

Results obtained by Grimble [Grimble,1991] using a *PI* controller are presented in Fig. 14. Table 3 summarizes the data, showing again clear advantages of controllers tuned by the *GA*.

Other system behaviour, such as no response oscillation or frequency domain restrictions can also be taken in account in *GA* tuned controllers. One has just to include new terms in the performance index, to impose system restrictions such as bandwidth, disturbance rejection or cut-off rate. For details, please refer to Risso [Risso, 2002].

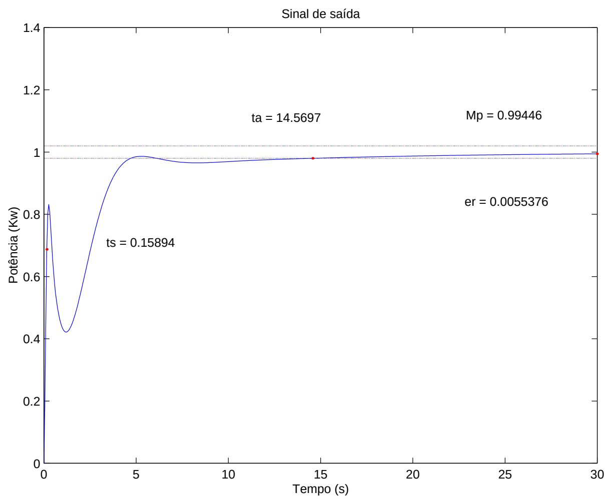

Figure 12: Case 2: step response GA run 2

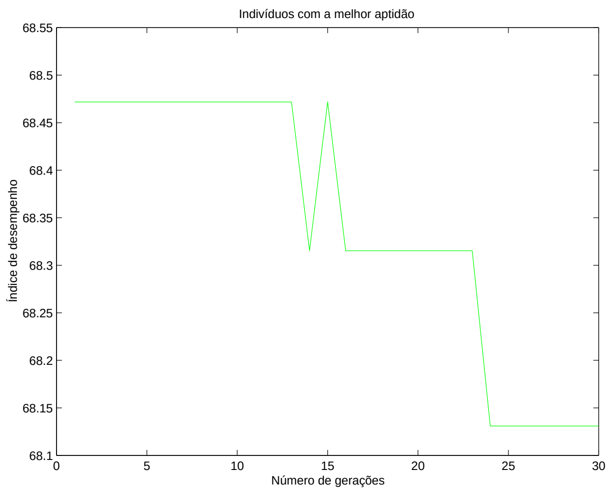

Figure 13: Case 2: GA convergence run 2

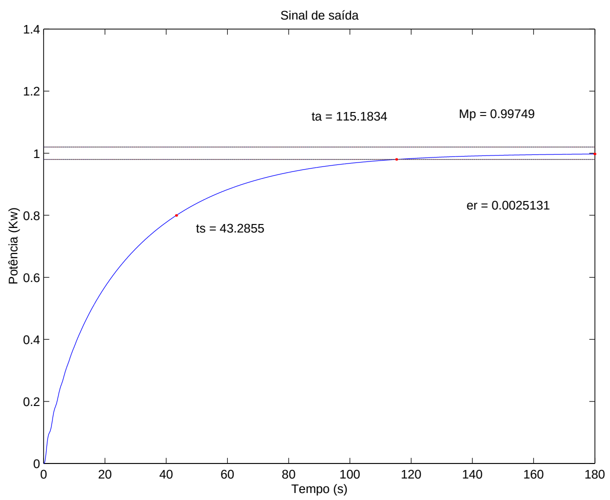

Figure 14: Case 2: PI controller by Grimble et al.

Table 3: Case 2 comparison

Controller	*PI*	*GA 1*	*GA 2*
$t_r(s)$	43.2	0.2	0.15
$t_s(s)$	115.2	16.9	14.5
M_p	0.0	0.0	0.0
$e_{ss}(\%)$	0.25	0.68	0.55
J	–	68.2	68.1

5. Conclusions

A procedure to tune *PID* controllers aiming the optimization of desired characteristics response was sucessfuly developped. The method was validated and also compared to other optimization approaches, showing similar or better performance, with the additional advantage of simple and robust algorithms. A disadvantage of the proposed method is the necessity of the definition of parameters for a performance index by the user, which impedes the procedure to be fully automatic. Actually, that definiton may sometimes turn to be laborius.

Some modifications of the method were already tryed, such as the use of decimal instead of binary strings, but this did not lead to consistent enhancements and were omitted in this work.

It seems to be easy to adapte the method presented here to tune other controller types, where some optimization is involved, such LQ, LQG or robust $\mathscr{H}_\infty$ controllers, when weighting parameters or weighting functions can be searched for and this will be the next step in this research.

6. Acknowledgments

The second author is grateful for the financial support from CAPES.

7. References

Anderson, C.; Leithead, W. E., Wilkie, J. Modelling of wind turbines by simple models. Wind Engineering, Brentwood, Essex, United Kingdom, v.14, n.4, p.247 – 274, 1990.

Anderson, C.; Leithead, W. E., Wilkie, J. Simulation of wind turbines by simple models. Wind Engineering, Brentwood, Essex, United Kingdom, v.1, p.336 – 340, 1991.

Aström, Karl J.; Haglund, Tore. Automatic tuning of PID Controllers. Research Triangle Park, NC., Instrument Society of America, 1988. 141p.

Cao, Y. J. ; Wu, Q. H. Teaching genetic algoritthm using Matlab. International Journal Engineering Education, Manchester v. 36, n.2, p.139 – 153, 1999.

De La Salle, S.A.; Reardon, D.; Leithead, W.E., et al. Review of wind turbine control. International Journal Control, Basingstoke, United Kingdom,v.52, n.6, p.1295 – 1310, dec., 1990.

Dorf, R. C. and Bishop, R. H. Modern control systems. 6.ed. Reading: Addison-Wesley, 1993. 717p.

Goldberg, David E. Genetic algorithms in search, otimization, and machine learning. Reading: Addison-Wesley, 1989. 412p.

Grimble, M. J.; Leithead, W. E.; Reardon, D.; De La Salle, S.A. Classical control of active pitch regulation of constant speed horizontal axis wind turbines. International journal control, Basingstoke, Hants, v.55, n.4, p.845 – 876, 1992.

Grimble, M. J.; Leithead, W. E.; Reardon, D.; De La Salle, S.A. Role and objectives of control for wind turbines. IEEE Proceedings: Generation Transmission and Distribution, Stevenage, Herts, v.138, n.2, pt. C., p.135 – 148, march, 1991.

Risso, G. Sintonia de Controladores PID por Algoritmos Genéticos - Dissertação de Mestrado EPUSP, 2002.

Holland, John Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence. Cambridge: Cambridge University Press, 1992. 211p.

Holland, John. Genetic algorithms. Scientific American, New York, v.267, n. 1, p.44 – 50, Julho, 1992.

KrishnaKumar, K. Genetic Algorithms: theory and control applications. In: Workshop, United Kingdom, August, 1994.

Mitchel; Melanie. An introduction to genetic algorithms. S.l.: MIT Press, 1999.

Novac, P.; Ekelund, T., Jovik, I., Schmidbauer, B. Modeling and control of variable-speed wind-turbine drive-system dynamics. IEEE Control Systems, p.28 – 38, August, 1995.

Spera, David A. Wind turbine technology: fundamental concepts of wind turbine engineering. New York : ASME Press, 1994. 638p.