Controller Tuning Using Genetic Algorithms

Ecaterina Emilia Vladu*, Toma Leonida Dragomir**

* Department of Electrotechnics and Informatics, University of Oradea, Oradea, Str. Armatei Române 5,

RO-410087, România, Phone: +40259408435, Fax: +40259408408,

e-mail: evladu@rdslink.roAuthor's Address, e-mail

** Department of Automation and Industrial Informatics, "Politehnica" University of Timisoara, Timişoara, Bd. Vasile Pârvan 2,

RO-300223, România, Phone: +40256403222, Fax: +4056403214,

e-mail: dragomir@aut.utt.ro

Abstract: *The increasing complexity of the modern control systems has emphasized the idea of applying new approaches in order to solve design problems for different control engineering problems. The genetic algorithms (GA) used in the paper are a promising heuristic approach to locating near-optimal tuning solutions. They are easy to implement and robust. This paper attempts to show how GAs can be applied for a two-step design problem. In the first step the GA assist the design of an ordinary PI controller for a plant with time delay, whereas in the second step GA help to tune the compensating controller for a structure with reference model. The presented cases study confirms that good performance can be achieved by the proposed method.*

Keywords: controller tuning, evolutionary techniques, genetic algorithms, reference model.

1 Introduction

The family of PID controllers represent, due to their simple structural construction the basic building blocks available in many process control systems. There are several classical methods for tuning a controller [1], [2], [3]. These methods have, however, some disadvantages: i) they are difficult to apply when the process is too complex to obtain precise dynamics of a plant; ii) they do not perform well for multiple specification design problems, such as the case required the optimisation of both reference response and disturbance response at the same time; iii) in most cases, the achievement of good quality performances leads to contradictory solutions. In practice the theoretical tuning methods are frequently substitute for empirical methods such: trial and error, continuous cycling method (Ziegler Nichols method), process reaction curve methods (Ziegler-Nichols and Cohen-Coon methods), and ITAE criteria. Despite their wide spread use and considerable history, the PID controller tuning is still an active area of research.

It is well known that by the modelling of dynamical systems it is practically impossible to describe the real plant exactly. Therefore the real system and the model of the control system used in design are not equal and will have a difference in their behaviour. The problem is that the controller gets designed on the base of a plant model, here named as the nominal model P_n, derived mathematically or identified, but is applied onto the real plant, which is a little bit different and whose real model P_r is actual an unknown system.

The controller found by the nominal model P_n is called the nominal controller C_n and CS_n means the nominal control system. Practically described, the nominal controller C_n, designed with the nominal model P_n, is applied to the real plant P_r. This system is called the real control system CS_r. The nominal system CS_n uses the same controller as in CS_r, but with the nominal model M_n.

In order to react on a deviation between the nominal model and the real model, the output of CS_n and CS_r gets coupled. A possible solution for increasing the robustness of dynamical systems is the feedback of coupling difference y_δ. The difference will counteract to the inaccuracy between the nominal and the real model. The approach is discussed in the recent paper [4] and its principle is shown in figure 1. The correction controller Cδ introduces the variable w_δ, which is the result of processing of the inaccuracy measure. The new reference value of the real system S_r gets the value:

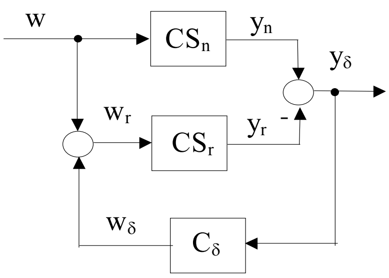


Figure 1 : Feedback of the difference between the nominal and real system.

$$w_r = w + w_\delta \qquad (1)$$

GAs offer a manner of solving optimisation problems, in particular the problems of controllers synthesis, which, thanks the categories of freedom which appears in the design can always be stated as optimisation problems [5], [6], [7]. Particularly, in the context of this paper, GAs can be used to the nominal and also to the correction controller's synthesis, C_n and respectively C_δ. The goal of this paper is to present the sides of principle and the results obtained in a study case. There are considered an evolutionary approach for PI and for the correction controllers tuning, using GA implemented in Matlab, thus allowing the use of all facilities offered by this environment, especially by Simulink [8]. The closed loop system's response, obtained by simulation, is associated with performance criteria types as: steady-state error, overshoot time, rise time, system time response, gain and phase margin etc. The controller's design can be formulated like an optimisation of both reference response and disturbance response at the same time, related to the minimization of an error that influence the mentioned performances.

The reminder of this paper is organized as follow: in Section 2 some basic aspects regarding the genetic algorithms are presented. The implementation details about the two-step design problem are developed in the third section. Section 4 concludes this paper.

2 Genetic algorithms

GAs are a powerful search algorithm that performs an exploration of the search space that evolves in analogy to the evolution in nature. The power of GAs consists in only needing objective function evaluations instead of derivatives or other auxiliary knowledge, to carry out their search. They use probabilistic transition rules instead of deterministic rules, and handle a population of candidate solutions (called individuals or chromosomes) that evolves iteratively. Each iteration of the algorithm is called generation. The evolution of the species is simulated through a fitness function and some genetic operators such as reproduction, crossover and mutation.

The fittest individuals will survive generation after generation while also reproducing and generating offspring's that might be stronger and stronger. At the same time, the weakest individuals disappear from each generation.

Individuals must be encoded in some alphabet, like binary strings, real numbers, vectors and other.

In a practical application of genetic algorithms, a population pool of chromosomes has to be installed and they can be randomly set initially. In each cycle of genetic evolution, a subsequent generation is created from the chromosomes in the current population. The cycle of evolution is repeated until a termination criterion is reached. The number of evolution cycles, or a predefined fitness value can set this criterion.

In essence, the procedure of a GA is given as follows:

1. *Generate randomly a population of chromosomes.*
2. *Calculate the fitness for each chromosome in the population.*
3. *Create offspring's by using genetic operators.*
4. *Stop if the search goal is achieved. Otherwise continue with Step 2.*

In this paper, the GA implemented in Matlab has the following main features:

- Individuals can be encoded like binary strings, real numbers, and vectors of binary strings or of real numbers, or like permutation strings;
- The genetic operators are implemented according with the encoding scheme used;
- Randomly generates the initial population, but allows the use of an initial population specified by the user;
- Performs the GA specific iteration;
- Includes the best performing individual of the parent generation in the new generation in order to prevent a good individual being lost by the probabilistic nature of reproduction;
- Allows the user to establish the GA parameters: the size of the population, the type of selection scheme, crossover and mutation and the probability of applying the genetic operators.

The GA is composed of two main components, which are strongly related to the problem to solve: the encoding scheme and the evaluation function.

The encoding scheme is used to represent the possible solutions to the problem. In this paper, the individuals are encoded as vectors of real numbers. The vector components are the controller's parameters to be found in the search process.

The evaluation function measures the quality of a particular solution. Each chromosome is associated with a fitness value, which in this case is the performance of the controller represented by the given chromosome. The fitness of a candidate is calculated here based on its simulated performance. Section 3 gives details about the design problem.

3 Controller Tuning Using GA

In the considered design problem, the plant P_n and the controller type must be known. Solving the problem consists in a two steps design: i) in tuning the parameters of a nominal C_n controller (proportional gain K_P, integral time T_I, derivative time T_D a.s.o.), ii) in tuning of the correction controller's C_δ (the gains K_1, K_2 a.s.o.) by using GAs. An individual *i* in GA is a row-vector and each element of the vector encodes one of the variables of that individual: K_P, T_I and T_D or other coefficients, as depicted in figure 2.

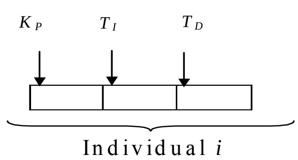


Figure : 2. Encoding scheme for an individual *i*

3.1 Design of the nominal controller C_n

The closed loop system's scheme considered in this paper is showed in figure 3, where P is the plant, and C the controller.

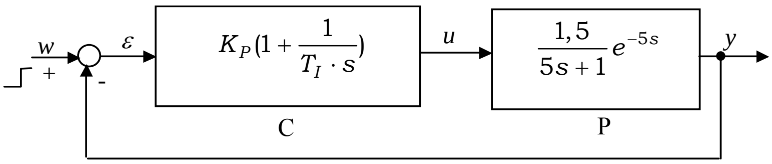


Figure 3 : The closed loop system used in case study.

The problem of tuning the controller of this system by using GAs, can be considered, in its turn, like a sequential control problem, represented in the block diagram in figure 4. GA acts as a controller who modifies the set of parameters $\{K_{Pi}\}_{i\in 1;p}$, $\{T_{Ii}\}_{i\in 1;p}$ of the "i"-th population, which consists of p individuals of Control Systems $\{CS_i\}_{i\in 1;p}$. This cycle is repeated until a convergence criteria CC is met, the operation being completed with the extraction of the best individual of the population.

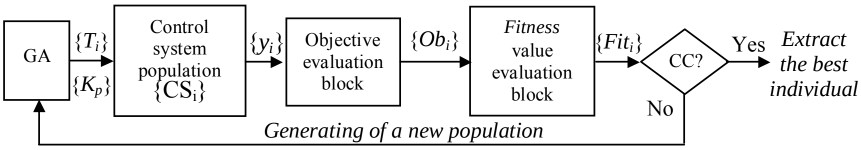


Figure 4 : Modelling of GA assisted design of the nominal control system

In the evaluation step of GAs, a simulation is performed for each CS_i. The desired performances as: steady-state error, overshoot time, rise time, system time response, gain and phase margin etc are associated to the shape of the system's transient response *y (t)*, obtained by simulation. The performance criteria are met if the system's response does not surpass an allowed area (the shaded zone) depicted in figure 5, where t_{max} is the maximum simulation time.

In this first step of the design, the performance criteria chose for an individual *CSi* was the area of the response curve that surpasses the allowed area in figure 5. Let be Ob_i its value (objective *i*):

$$Ob_i = c_1\int_0^{t_2}\{\max[y_i(t)-f_4)],0\}dt + c_2\int_{t_1}^{t_2}\{\max[f_1-y_i(t),0]dt + \\ + c_3\int_{t_2}^{t_{\max}}\{\max[y_i(t)-f_3],0\}dt + c_4\int_{t_2}^{t_{\max}}\{\max[f_2-y_i(t),0]dt \quad (2)$$

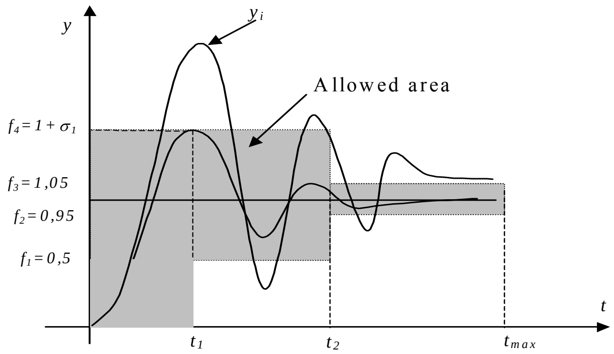


Figure 5: The allowed area for optimal parameters tuning

In order to assure better performances with respect to the system's response time, the objective in relation (2) was completed with an additional term, given by relation (3), which includes the derivative of the system's response *y (t)*:

$$-c_5 \int_0^{t_3} \dot{y}_i^2(t)dt \, . \qquad (3)$$

It has to be noted that the designer can chose appropriate values for the parameters f_1, f_2, f_3, f_4, c_1, c_2, c_3, c_4, c_5, $t1$, t_2, t_3 *and* t_{max} for the Ob_i expression, according with different practical constraints. Ob_i is to be minimized, so the fitness function Fit_i for an individual *i* becomes

$$Fit_i = \frac{1}{1 + Ob_i} \qquad (4)$$

While it is difficult to represent the objective function in a mathematical form it was implemented, in a real time manner, in the same Simulink model as the control system. First, the GA randomly generates an initial population of chromosomes. In the evaluation step, the algorithm calls Simulink by transmitting for each individual, a set of parameters. At the end of the simulation process, the Simulink model transmits to the GA the Fitness value, which is then used, until the CC block allows, in the other steps of the optimisation process.

By applying GA with f1 = 0,8; f_2 = 0,98; f_3 = 1,02; f_4 = 1,1 and the weighting factors c_1=c_2=c_3=c_4= c_5=0,2, the first five best solutions given in Table 1 were obtained. Figure 6a depicts the system's response by using the solution number 1 from the table, while the figures 6b and 6c use less performing obtained individuals during the "GA assisted design": K_P= 0,0716, T_I = 1,0000, with *Fit* = 0,7501 respectively K_P= 0,1043, T_I = 2,167, with *Fit* = 0,9407. Figure 6d depicts the average fitness *M* and the standard deviation *S* of the fitness along the N = 30 generations of the algorithm.

Table 1. The solutions in the first step of the study case

Solution number	K_P	T_I [sec.]	*Fitness*	Parameters of GA
1	0.56400	6.34020	1.0860	Population size: N = 30
2	0.56395	6.33690	1.0858	Tournament selection, T=5
3	0.56400	6.34300	1.0858	Arithmetic crossover, Pc= 0.6
4	0.56400	6.34200	1.0855	Uniform mutation, Pm=0.1
5	0.56399	6.33850	1.0855	Stopping criteria: 30 generations

$$M = \frac{1}{N} \cdot \sum_{i=1}^{N} Fit_i \ , \ S = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (Fit_i - M)^2} \ . \tag{5}$$

By examining the obtained results, can be seen that, after 15 generations, the GA performs a local search and the fitness value and the average fitness *M* increase slowly till the end of the evolution. The standard deviation *S* increases till generation 10. After that, it decreases till the end of the evolution.

This step of the design was further studied by modifying the parameters of the GA and the performance criteria. The obtained results conclude that reducing the population size and the number of generations can be also solving the problem. Also, by adding more adequate terms to relation (2), the quality of the solutions increases.

3.2 Design of the correction controller C_δ

In the second design stage, the control structure depicted in figure 1 was considered, in which the nominal control system which was previously designed, with *Kp*=0.56399 and *Ti*=6.33850, has the role of the nominal control system CS_n.

For the correcting block two types of rational structures were considered:

$$H_{\delta 1}(s) = \frac{K_1 \cdot s^2 + K_2 \cdot s + K_3}{s^2 + K_4 \cdot s + K_5} \ , \ H_{\delta 2}(s) = \frac{K_1 \cdot s + 0.01}{s + K_2} \ . \tag{6}$$

The case considered consist in admitting that the parameters of the real plant differ from their nominal values, $H_{\Pr}(s) = \frac{1,4}{5,2s+1} \cdot e^{-5,4s}$ versus $H_{Pn}(s) = \frac{1,5}{5s+1} e^{-5s}$.

The correction blocks were subjects in a design by GA according with the principle represented on Figure 4. Consequently, the required parameters corresponded in series to the sets of {K_1, K_5} respectively {K_1, K_2}. The minimization criteria took into account the difference for a step signal response:

$$Ob_i = \int_0^{t_3} \left| y_{\delta i}(t) \right| dt \text{, where } y_\delta = y_n\text{-}y_r \tag{7}$$

The best-obtained results were the following (included in parentheses are the GA's parameters):

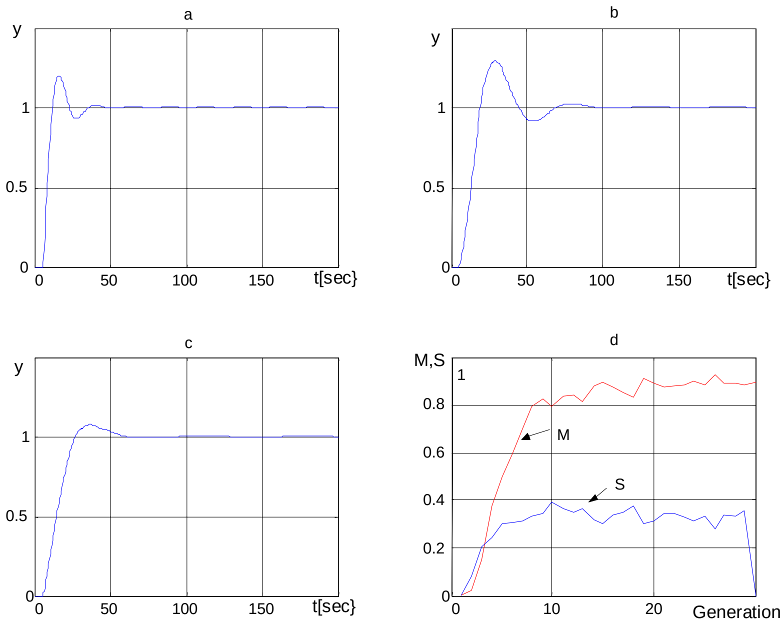


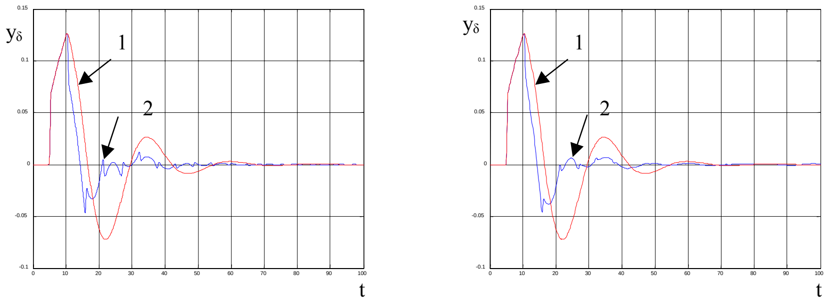
Figure 6 : The behaviour of some individuals CS

- For the correcting block with the structure: $H_{\delta 1}(s)$: K_1 = 83.6160, K_2 = 79.4398, K_3 = 0, K_4 = 18.2937, K_5 = 20.8571, with *Fit max*= 0.5085.
- For the correcting block with the structure: $H_{\delta 2}(s)$: K_1 = 86.1877, K_2 = 22.5174, with *Fit max*= 0.5019.

The variation of the difference y_δ in the case that do not corrects the real system's behaviour was denoted in figure 7 with 1, while with 2 the difference obtained by using the scheme of figure 1. The results obtained corresponding to the two types of correcting elements are very closed and consist of accelerating the amortising process of the difference between the nominal one and the true system behaviour.

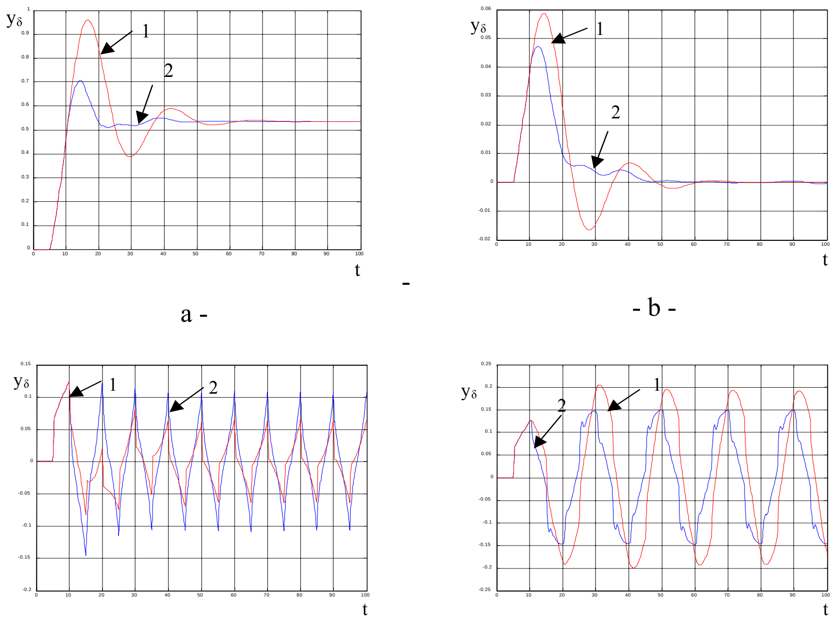
Since the conceived system is designed to function with various reference signals, its behaviour with regard to ramp impulse (figure 8 a), exponential (figure 8 b), and rectangular with the period of 10 sec (figure 8 c), and 20 sec (figure 8 d), was further examined. Figure 8 presents the obtained results in the case $\dfrac{K_1 \cdot s + 0.01}{s + K_2}$.

It can be seen that, excepting the case c, the obtained results with the control structure depicted in figure 1 (the curves denoted with 2) are better. The behaviour in the first 10 seconds is due to the time delay.



- a - - b -

Figure 7 : The behaviour of the structures with automatic correction with respect to a step reference signal



- c - - d -

Figure. 8 : Comparing the automatic correcting structure performances with respect to different reference signals.

4. Conclusions

The design of the control systems by using GAs is a method that can help the designer in many respects:

i) operating with a reduced number of design methods to establish the type of the controller;

ii) ii) possibility of easily configuring the dynamic behaviour of the control system;

iii) iii) starting the design with a reduced amount of information about the controller (type and allowable range of the parameters), but keeping sight of the behaviour of the control system.

In this work is given point of these features by considering the problem of designing a control system for a plant of first order with time delay. In the design, a working scheme based on a compensation of the uncertainty of the parameters of the controlled plant was studied. The compensating process is based on the use as reference model of the nominal model.

Firstly, this work has a methodological relevance. Secondly, it reveals the usefulness of the compensation principle. In the absence of methodology of design of the compensation blocks, GAs offers an undeniable alternative in the design, which uses this principle.

References

[1] Föllinger, O.: Regelungstechnik, Hüthig, 1994.

[2] Levine, W.S. (editor): Control Systems Fundamentals, CRC Press, 2000.

[3] Åström, K. J., Wittemark, B.: Computer-Controlled Systems, Prentice Hall, 1993.

[4] Dragomir, T.L., Roth, H., Roesch, O.J.: Control Loop Adjustment of plant Uncertainties, Periodica Politehnica, Transactions on Automatic Control and Computer Science, Vol.49 (63), 2004 (to be published)

[5] Vladu, E.E.: Contributions in using genetic algorithms in engineering, PhD Thesis, “Politehnica” University of Timisoara, 2003.

[6] Gen, M., Cheng, R.: Genetic Algorithms and engineering design, Wiley series in engineering design and automation New-York 1997.

[7] Goldberg, D.E.: Genetic Algorithms in Search, Optimisation and Machine Learning. Addison –Wesley Publishing Company, Inc., Massachusetts, 1989

[8] Math Works Inc. "Simulink Toolbox for use with Matlab" User's guide Version 2 2000