Research Journal of Recent Sciences ISSN 2277-2502
Vol. **5(1),** 16-26, January **(2016)** *Res.J.Recent Sci.*

Open Loop Step test used for Process Identification and PID tuning controller by Genetic Algorithms

Lagunas Jiménez Rubén[1]*, Lorandi Medina Alberto[2] and Hermida Saba Guillermo[2]
[1]Universidad Autonoma de Campeche, MÉXICO
[2]Instituto de Ingenieria, Universidad Veracruzana, MEXICO
jrlaguna@uacam.mx

Available online at: www.isca.in, www.isca.me
Received 12th August 2014, revised 31st December 2014, accepted 6th September 2015

Abstract

In this article a proposal to solve two control problems from multiple point identification process frequency response of linear models, using an open loop step, is presented. The identified points are used, in one case a PID controller tuning, and the other application deals with transfer function modeling problem, both problems are stated as a nonlinear least squares unconstrained minimization problem. The optimization problem is solved with a simple genetic algorithm.

Keywords: FFT, modeling, nonlinear least squares optimization, PID controller, genetic algorithm.

Introduction

Proportional-Integral-Derivative (PID) controllers are commonly used in process control systems. In process control, more than ninety-five percent of the control loops are of PI or PID type[1,2]. Since Ziegler and Nichols[3], proposed their empirical method to tune PID controllers, to date, many relevant methods to improve the tuning of PID controllers has been reported at the control literature, one of them is a tutorial given by Hang et al.[4].

As is well known, the dynamics of a process can be known from the transient response, so when it gets the step response is possible to determine both the process gain and the process dynamics. Due to this statement, in this work, the frequency response is obtained from the step response in a open loop system. The size of the step can be as small as it has desired, this is a great advantage because it can apply a small step near the operation point, without significantly affecting process safety.

An open-loop step test is used when the process is at zero initial state or in a nonzero steady state, so that it can see the dynamic response of the system, for a step input, in order to realize a model identification or controller designing[5]. Several researchers have made important contributions on Control-oriented model identification methods[6-10]. A significant tutorial review on process identification from step or relay feedback was presented by Liu *et al*[5], in this work the most important methods for identification and controllers designing in last three decades are presented. In the first proposals on auto-tuning methods, one estimated point over Nyquist curve is enough to tune a PID controller. In recently studies, it has been shown that the multiple identified points allow better PID tuning controller[4,5]. This work presents two applications of the multiple-point identification method, in order to tune PID controllers and, on the other hand, to obtain transfer function coefficients. The control problem is posed as a nonlinear least squares unconstrained problem.

A genetic algorithm is proposed to solve the optimization problem. The same methodology can be used for both cases: PID tuning and transfer function modeling. Nonlinear least squares methods are important iterative procedures in order to reduce the sum of the squares of the errors between a proposed function and the measured data points. These kinds of problems are common when it wants to fit proposed functions from experimental date. The Levenberg-Marquardt algorithm[11,12], is the most common method for nonlinear least-squares minimization, nevertheless it can suffer from a slow convergence, and it is possible to finds only a local minimum[12].

The PID's designed with this method takes into account the effect of the sensitivity function values of the closed-loop system as a measure of robustness against possible variations in the parameters of the plant[1,2,13,14]. The proposed examples in this article cover a wide range of cases**:** stable, with short and long dead times, whit real and complex poles, and with positive and negative zeros, which are representative of the process control literature[4,13].

The contents of the paper are described as follows: In section 2 the basic definitions of a nonlinear least squares unconstrained minimization problem, use of open loop step transient test, as well as a simple genetic algorithm procedure are shown. In section 3 some examples of PID tuning controllers and system identification, based on multiple points, of the frequency response are presented. Conclusions are contained in section-4.

Basic Concepts

Unconstrained minimization problem: In a large number of practical problems, the objective function f (x) is a sum of squares of nonlinear functions

$$f(x) = \frac{1}{2}\sum_{j=1}^{m}(r_j(x))^2 = \frac{1}{2}\|r(x)\|_2^2 \quad (1)$$

That needs to be minimized. We consider the following problem

$$\min_x f(x) = \min_x \frac{1}{2}\sum_{j=1}^{m}(r_j(x))^2 \quad (2)$$

This is an unconstrained nonlinear least squares minimization problem. In this procedure the sum of squares of these functions is the quantity to be minimized. Such problems arise when we want to fit the function parameters from experimental data: if φ(x; t) represents the model function with *t* as an independent variable, then each $r_j(x) = \varphi(x; t_j) - y_j$, where $\varphi(t_j,y_j)$ is the given set of data points[11,12]. In this work an unconstrained nonlinear least-squares minimization problem is used.

Use of open loop step transient: The idea of identifying multiple points of the frequency response is presented by Wang et al.[7,15]. For an open loop step test, a schematic block system is shown in figure-2. The signal input *u(t)* and output *y(t)* are considered from the initial time until the steady state is reached, after the transient step response. *U(t)* and *y(t)*cannot be integrated because they do not reach zero value in a finite time (at T_{ss} time). Due to, they cannot be directly transformed to frequency response using *FFT*. To avoid this problem, an exponential function $e^{-\alpha t}$isused[7]

$$\tilde{u}(t) = u(t)e^{-\propto t} \quad (3)$$

and

$$\tilde{y}(t) = y(t)e^{-\propto t} \quad (4)$$

Thus signal *u(t)* and *y(t)* will tend to zero exponentially when *t* approaches to infinite value, as can be shown in figure-1. Applying the Fourier transform to (3) and (4) it has

$$\tilde{U}(t) = \int_0^\infty \tilde{u}(t)e^{-jwt}\,dt = U(jw+\propto)$$

and

$$\tilde{Y}(t) = \int_0^\infty \tilde{y}(t)e^{-jwt}\,dt = Y(jw+\propto)$$

For a transfer function *G(s)=Y(s)/U(s), at s=jw+α*, it has

$$G(jw+\propto) = \frac{Y(jw+\propto)}{u(jw+\propto)} = \frac{\tilde{Y}(jw)}{\tilde{U}(jw)} \quad (5)$$

Now the discrete values of $\tilde{Y}(jw)$ and $\tilde{U}(jw)$ can be determined using the standard *FFT* algorithm[7,15]. Thus, the shifted process frequency response *G (jw+α)* can be obtained from (5). However, if one want to obtain *G(jw)* from *G(jw+α)*, then it should first compute the inverse *FFT* of *G(jw+α)* as

$$\tilde{g}(kT) = FFT^{-1}\big(G(jw+\propto)\big) = g(kt)e^{-\propto kt}$$

Then, the process impulse response (kT) is

$$g(kT) = \tilde{g}(kT)e^{\propto kT}$$

Now, the *FFT* is again applied to g(kT) to obtain the process frequency response:

$$G(jw) = FFT(g(kt)) \quad (6)$$

In this identification problem is very important the adequate selection of α value, a rule to compute the α value in terms of the T_{ss} time (see figure-1) is proposed, where the system reaches a steady value, after the transient step response. The value of α, it can be computed by means of:

$$\propto < \frac{1}{Tss}\, ln\frac{\Delta y(Tss)}{\delta}$$

Where $\Delta y(T_{ss})=y(T_{ss})-y(0)$, means the transitory output response in terms of the settling time (T_{ss}) to the step change, in which *y(0)* indicates initial steady output value before the step test. Δ is a computational threshold, where for practical purposes, it can take values lower than $\Delta y(T_{ss})x10^{-6}$.

The method of open loop step test can accurately identify as many as desired frequency response points with one step experiment. In both applications: PID tuning and transfer function modeling, the shifted frequency response may be used without the needing to computer *G(jw)*. To illustrate the method, a model with oscillatory dynamics is proposed.

$$G(s) = \frac{1.25}{0.25s^2+.7s+1}e^{-.234s} \quad (7)$$

Figure-3 shows the identified multiple points for model with oscillatory dynamics using this method, for *G(jw)*.

And G *(jw+α)* plot, where *α=0.85,* is given by figure-4

Simple Genetic Algorithms: The genetic algorithm is a useful tool to solve both constrained and unconstrained optimization problems that takes principles of biological evolution[8,14,16-18]. At present work, each of the individuals in the population (chromosomes), contain the parameters included in the fitness function, as an example, in the process to tune the PID controller, each chromosome contains the coded parameters of the controller [*Kp, Ki, Kd*]. The following procedure summarizes the main steps that the genetic algorithm executes:

The algorithm starts by creating a random initial population.

The algorithm creates generations of new populations. At each step, the algorithm uses the individuals in the current generation to create the subsequent population. To generate the new population, the following steps are realized:

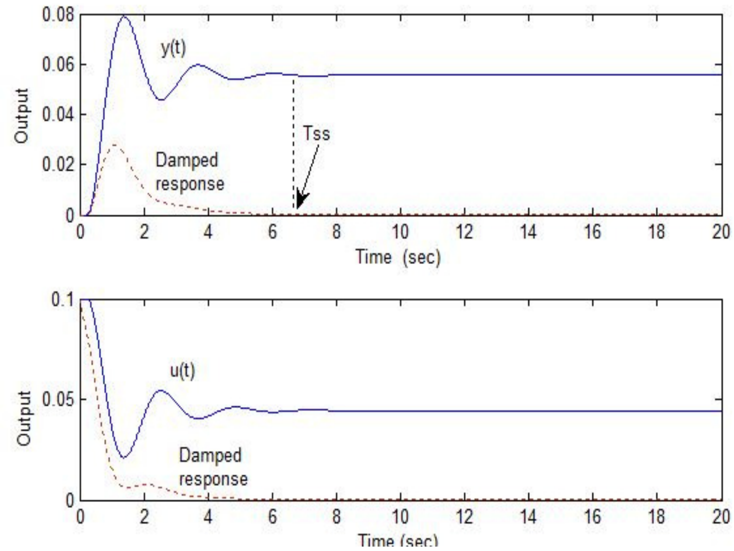

Figure-1
Signals under open loop step

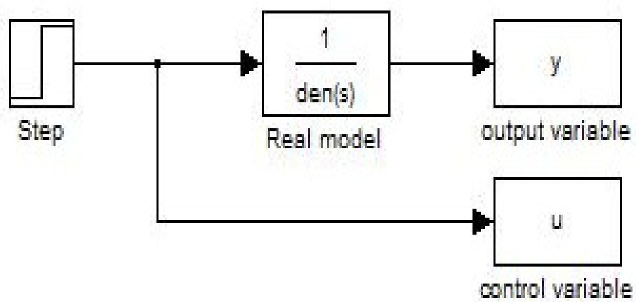

Figure-2
Schematic of open loop step test

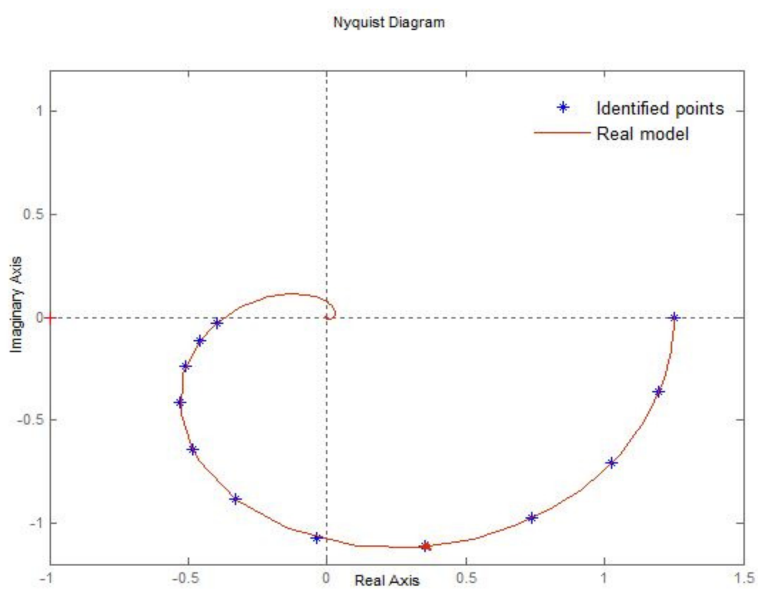

Figure-3
Nyquist plot for *G(jw)*

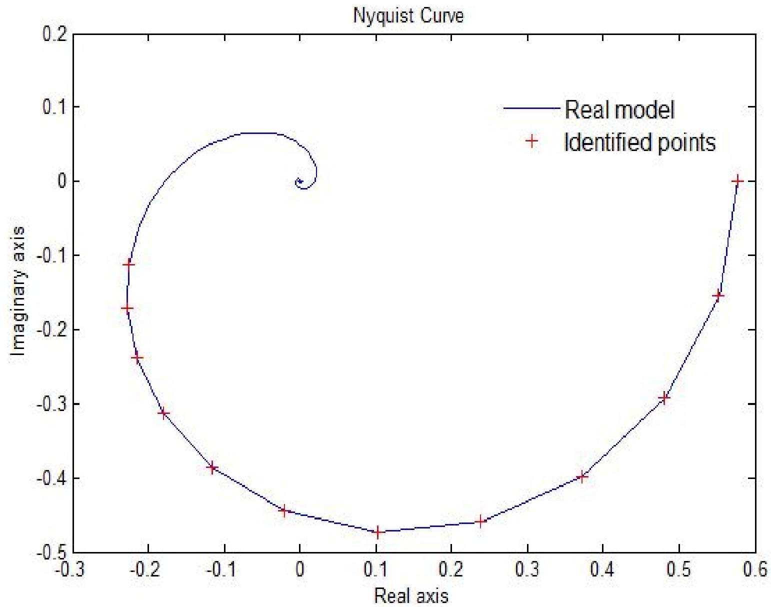

Figure-4
Nyquist plot for *G(jw+α)*

Assign a grade to each member of the current population by computing its fitness value. Selects members, called parents, based on their fitness. Some of the individuals in the current population that have better grades are chosen as elite. These elite individuals are passed on to the next population. Children from the parents are produced by means of crossover and mutation operators. The current population is replaced with the children to form the next generation. The algorithm terminates when some stopping criterion is met.

Applications

Designing of PID controllers: Designing of PID controllers by means of frequency response fitting has been reported in remarkable literature about processes control[4,15,19]. This method takes several points of the frequency response to shape the desired dynamics over a wide frequency range. Thus the closed-loop performance is better than in the case of only one or two points are taking to design PI or PID controllers. Supposing the points of G(jwi), i=1,2,…,m, are known. The performance of the control system can be expressed as a desirable closed loop transfer function

$$H_d(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n+\omega_n^2}e^{-Ls} \tag{8}$$

Where L is called as *apparent dead-time* of the process, w_n and ζ determine the behavior of the desired closed-loop response[4]. The control specifications can begivenin terms of phase margin Φ_m, and gain margin A_m. The value of w_n and ζin H_d are computed by[4]

$$\zeta = \sqrt{\frac{1-cos^2(\phi_m)}{4cos^2\phi_m}} \text{and} \omega_n = \frac{\tan^{-1}(\frac{2\zeta p}{p^2-1})}{pL}$$

Where p is the positive root of equation

$$(A_m-1)^2 = 4\zeta^2p^2 + (1-p^2)^2$$

Most accepted values for ζ and w_n L are 0.707 and 2, respectively, this means that overshoot value of the objective set-point step response is about 5%, the phase margin is 60° and the gain margin is 2.2[4]. The open-loop transfer function corresponding to G_d is

$$G_d = \frac{H_d}{1-H_d} \tag{9}$$

The controller *C(jw)* is designed such that the actual *GC(jw)* is fitted to the desired transfer function $G_d(jw)$, as much as possible. Then the close loop system will have the desired performance. The PID controller desired can be obtained by minimizing the objective function given from the sum of squared differences between computed and recorded frequency response points

$$CG(jw_i) = \frac{Kpjw_i+Ki+Kd(jw_i)^2}{jw_i}G(jw_i) \tag{10}$$

$$CG'(jw_i) = \begin{bmatrix} Real\big(GC(jw_i)\big) \\ Imag\big(GC(jw_i)\big) \end{bmatrix} \text{ and } G_d'(jw_i) = \begin{bmatrix} Real(G_d(jw_i)) \\ Imag(G_d(jw_i)) \end{bmatrix}$$

The objective function

$$y = \sum_1^m \left| CG'(jw_i) - G_d'(jw_i) \right|^2 \qquad (11)$$

If the PID controller is designed from *G (jw+α)*, then

$$C(jw_i + \alpha) = \frac{Kp(jw_i+\alpha)+Ki+Kd(jw_i+\alpha)^2}{(jw_i+\alpha)} \qquad (12)$$

$$CG(jw_i + \alpha) = C(jw_i + \alpha)G(jw_i + \alpha)$$

$$CG'(jw_i + \alpha) = \begin{bmatrix} Real\big(GC(jw_i + \alpha)\big) \\ Imag\big(GC(jw_i + \alpha)\big) \end{bmatrix} \text{and} G_d'(jw_i + \alpha) = \begin{bmatrix} Real(G_d(jw_i + \alpha)) \\ Imag(G_d(jw_i + \alpha)) \end{bmatrix}$$

The objective function

$$y = \sum_1^m \left| CG'(jw_i + \alpha) - G_d'(jw_i + \alpha) \right|^2 \qquad (13)$$

The solution of the problem is obtained by minimizing *y*.

In this work the identified points were obtained from a schematic Simulink® system where the system feedback is simulated**.** To solve the optimization problem**,** the MATLAB® Genetic Algorithm Optimizations Using the Optimization Tool GUI is used.

Example-1: Third order plus dead time system.

The designed PID is solved by minimizing the equation 11 by means of a simple genetic algorithm. The PID parameters are coded and arranged into each individual (chromosome), of population in the genetic process. Multiple points are from *G(jw)*

$$G(s) = \frac{1}{(s+1)^3} e^{-5s} \qquad (14)$$

The apparent dead-time *L=4.5.* The designed PID controller is

$$C(s) = (0.4318 + \frac{0.111}{s} + 0.443s) \qquad (15)$$

The feedback system response with the PID controller designed is shown in figure-5.

Example-2: Consider a model with oscillatory dynamics

$$G(s) = \frac{1.25}{0.25s^2 + .7s + 1} e^{-.234s} \qquad (16)$$

The identified points for this model are showed in figure-3-4. In this example the apparent dead-time value *L=0.23,* is proposed.

The designed PID is solved by minimizing the equation 11 by means of a simple genetic algorithm.

$$C(s) = (1.453 + \frac{2}{s} + 0.561s] \qquad (17)$$

And from *G(jw+α)*, the tuned PID controller is

$$C(s) = (1.45 + \frac{2}{s} + 0.561s] \qquad (18)$$

Equation-15-16 show that both PID's controllers have very close values as might be expected.

Genetic algorithms give the ending fitness value and the PID parameters as is shown in figure-6.

Performance of the PID designed is shown in the figure-7. The time response shows that the overshoot value is close of 5%, as it was proposed.

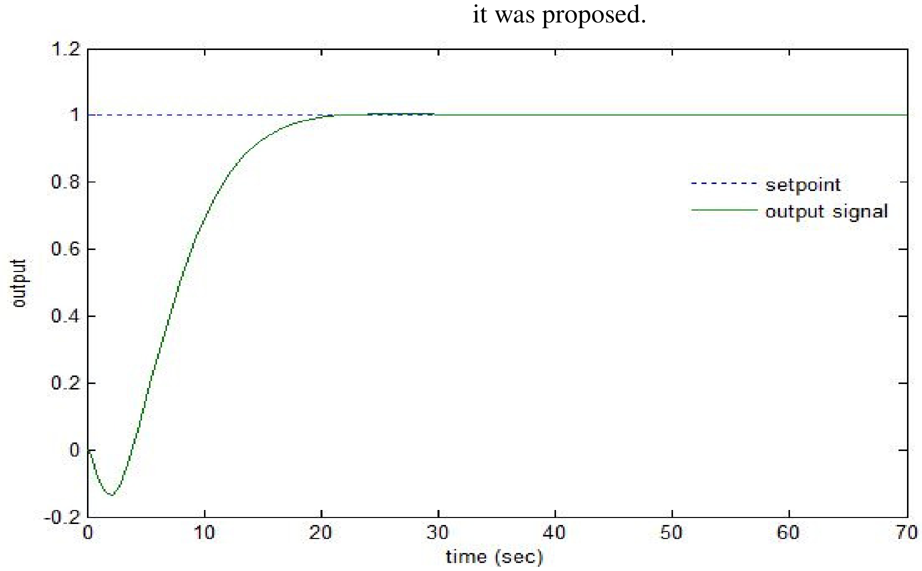

Figure-5
Close loop response of example-1

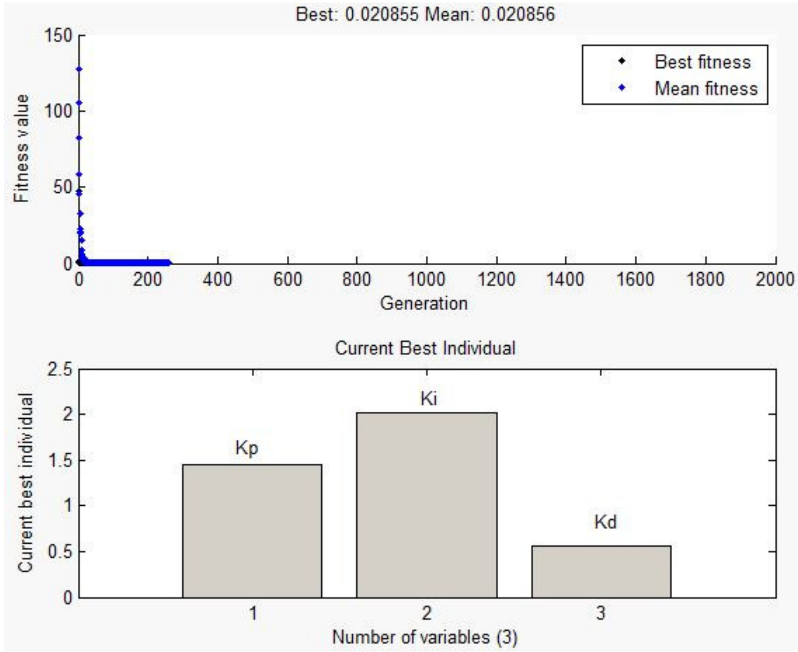

Figure-6
Ending fitness value and PID parameters for an oscillatory process

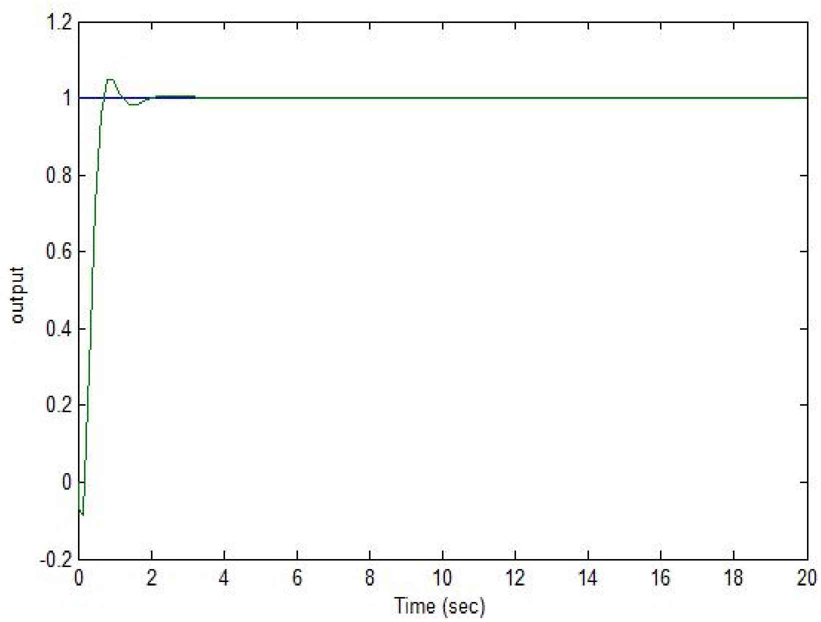

Figure-7
Control performance for an oscillatory process

Example-3: Considerer a third order model

$$G(s) = \frac{1}{(s+1)^3} \tag{19}$$

For this model the value of *apparent dead-time* of the process *L=0.3* was proposed. The modeling error for this example was 0.08%.

The design PID controller is

$$C(s) = (5.1927 + \frac{1.97}{s} + 4.83s) \tag{20}$$

Performance of the PID designed is shown in the figure-8.

The sensitivity to modeling errors: Since the controller is tuned for a particular process, it is desirable that the closed loop system is not very sensitive to variations of the process dynamics. A convenient way to express the sensitivity of the closed loop system is through the sensitivity function *S(s), defined as:* $S(s) = \frac{1}{1+L(s)}$, where *L(s)* denotes the loop transfer function[12,14,19-21]. *L (s)* is given by:

$$L(s) = C(s)G(s) = G(s)k\left(1 + \frac{1}{T_i s} + T_d s\right)$$

The maximum sensitivity (frequency response) is then given by $M_s = \max_{\omega}|S(i\omega)|$. Therefore M_s is given by $M_s = \| S(s) \|_{\infty}$. On the other hand, it is known that the quantity M_s, is the inverse of the shortest distance from the Nyquist curve of loop transfer function to the critical point s=-1. Typical values of M_s are in the range from *1.2 to 2.0*[13].

Table 1 shows the values of M_s, A_m and Φ_m for the three presented examples.

The operation of genetic algorithm was configured with the following parameter values: Population size: 100. Stochastic uniform Selection. Crossover function: Scattered. Mutation function: Gaussian. Number of iterations: 500. Crossover probability: 0.8. Mutation Probability: 0.09. Elite count: 2.

Transfer Function modeling: A mathematical model is necessary in many applications of automatic control. In this work a second order plus dead-time model is proposed. The identification at models with dead-time is usually a non-linear problem[4,8,20]. This characteristic presents a good opportunity to apply a genetic algorithm to solve the optimization problem.

$$G(s) = \frac{1}{as^2+bs+c}e^{-Ls} \tag{21}$$

This second-order-plus-dead-time model can represent both monotonic and oscillatory processes.

Transfer function modeling from *G(jw):* Suppose the process frequency response $G(jw_i)$, $i=1,2,\ldots,M$ is presented, because of they are required to be fitted into G(s) in (21) such that

$$G_m(j\omega_i) = \frac{1}{a(j\omega_i)^2+bj\omega_i+c}e^{-L\omega_i} \tag{22}$$

Where $i=1,2,\ldots, M$
then

$$G_m{}'(jw_i) = \begin{bmatrix} Real\big(G_m(jw_i)\big) \\ Imag\big(G_m(jw_i)\big) \end{bmatrix}$$

And the identified points of G(jw)

$$G'(jw_i) = \begin{bmatrix} Real(G(jw_i)) \\ Imag(G(jw_i)) \end{bmatrix}$$

The objective function is

$$y = \sum_1^m \left|G_m'(jw_i) - G'(jw_i)\right|^2 \tag{23}$$

The solution of the problem is obtained by

$$\min_i \sum_1^m \left|G_m'(jw_i) - G'(jw_i)\right|^2 \tag{24}$$

Transfer function modeling from *G (jw+α):* Suppose the shifted frequency response of the process $G(jw_i+\alpha)$, $i=1,2,\ldots,M$ is available, then the model given by equation-21 is used, such that

$$G_m(j\omega_i+\propto) = \frac{1}{a(j\omega_i+\propto)^2+b(j\omega_i+\propto)+c}e^{-L(j\omega_i+\propto)} \tag{25}$$

Where $i=1,2,\ldots,M$

$$\text{then } G_m{}'(jw_i+\alpha) = \begin{bmatrix} Real\big(G_m(jw_i+\alpha)\big) \\ Imag\big(G_m(jw_i+\alpha)\big) \end{bmatrix}$$

And the identified points of G(jw+α)

$$G'(jw_i+\alpha) = \begin{bmatrix} Real(G(jw_i+\alpha)) \\ Imag(G(jw_i+\alpha)) \end{bmatrix}$$

The objective function is

$$y = \sum_1^m \left|G_m'(jw_i+\alpha) - G'(jw_i+\alpha)\right|^2 \tag{26}$$

The solution of the problem is obtained by

$$\min_i \sum_1^m \left|G_m'(jw_i+\alpha) - G'(jw_i+\alpha)\right|^2 \tag{27}$$

In order to illustrate the application of this method, the following examples were proposed to obtain the identified models from Multiple points from G (jw) and G (jw+α). The estimated models were solved by minimizing the equations-24 and 27 by means of a simple genetic algorithm.

Example-4: Considerer a third order plus dead time system

$$G(s) = \frac{1}{(s+1)(5s+1)^2}e^{-2.5s} \tag{28}$$

The Identified model parameters [a, b, c, L], are coded and arranged into each individual (chromosome), of population in the genetic process. The value of modeling error is 0.0154%

The identified SOPDT from *G(jw)*is

$$\hat{G}(s) = \frac{1}{27.12s^2+10.19s+1.002}e^{-3.3s} \tag{29}$$

The real model, identified points and identified model are shown in figure-9.

Table-1
Values of *Ms,* A_m and Φ_m

Model	*Ms*	*Gain margin*	*Phase margin*
$\frac{1}{(s+1)^3}e^{-5s}$	*1.547*	*3.16*	*63.74*
$\frac{1.25}{0.25s^2+.7s+1}e^{-.234s}$	1.685	3.13	59.4°
$\frac{1}{(s+1)^3}$	1.52	2.88	63.74°

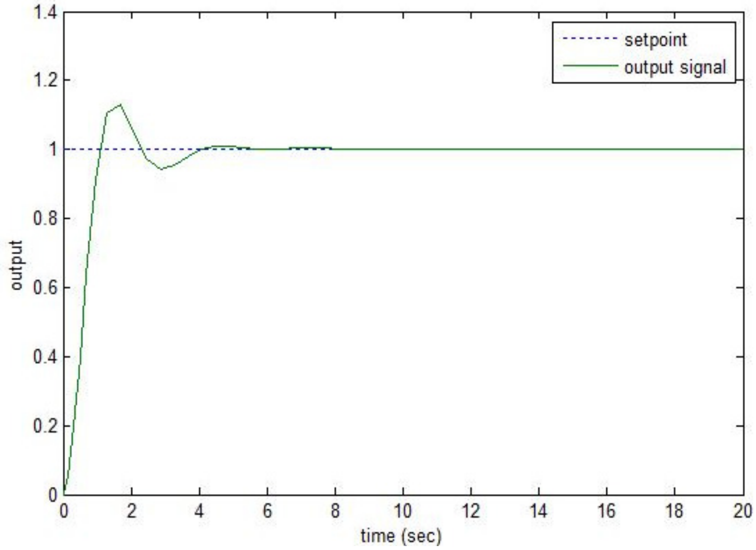

Figure-8
Control performance for third order model process

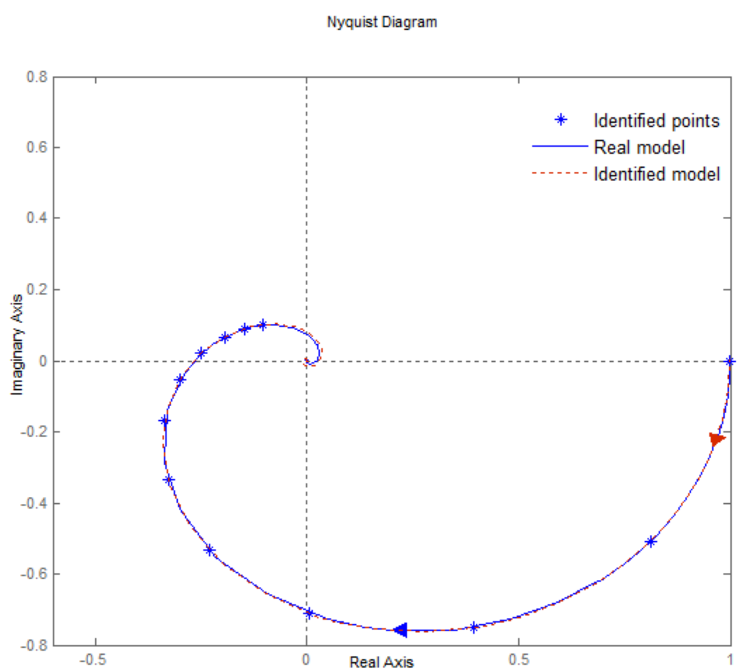

Figure-9
Identified model and real model for example 4

Example-5: Third order plus dead time system

$$G(s) = \frac{1}{(s+1)^3} e^{-5s} \quad (30)$$

In figure-10 the final values of the fitness function and the identified model parameters are presented.

Thus the identified model was attained from *G(jw)*. Model error =0.0152% and α=0.24.

$$\frac{1}{1.888s^2+2.59s+1} e^{-5.408s} \quad (31)$$

Real and identified models are shown in figure-11

Transfer function modeling for Processes with long dead-time: Processes with long dead-time are common in most of the industrial processes and can be satisfactorily approximated by a structure in form of

$$G(s) = \frac{K}{(Ts+1)^2} e^{-Ls} \quad (32)$$

Example-6: Consider the following long dead-time process[4]

$$G(s) = \frac{0.57}{(8.6s+1)^2} e^{-18.7s} \quad (33)$$

The identified model is given in equation-34. It was obtained by the same method as was presented previously, with α=0.08. Figure-12 shows the identified points on the Nyquist curve from *G (jw+α)*. Identification error=0.0056%

$$\hat{G}(s) = \frac{0.5703}{(8.606s+1)^2} e^{-18.7s} \quad (34)$$

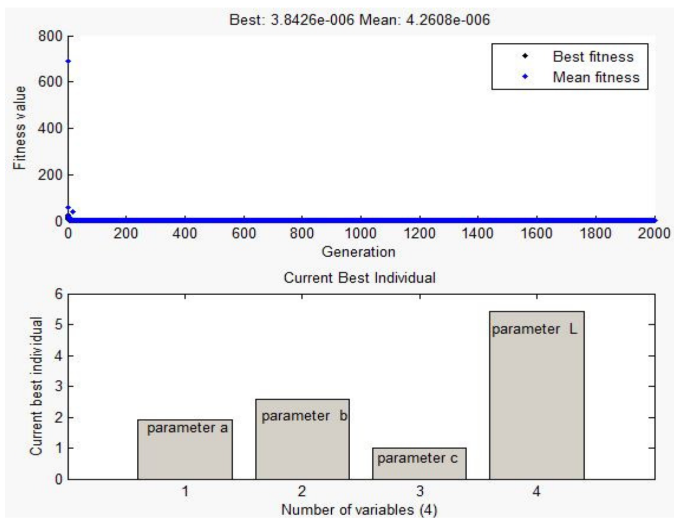

Figure-10
Fitness function and model parameters of a third order plus dead time system

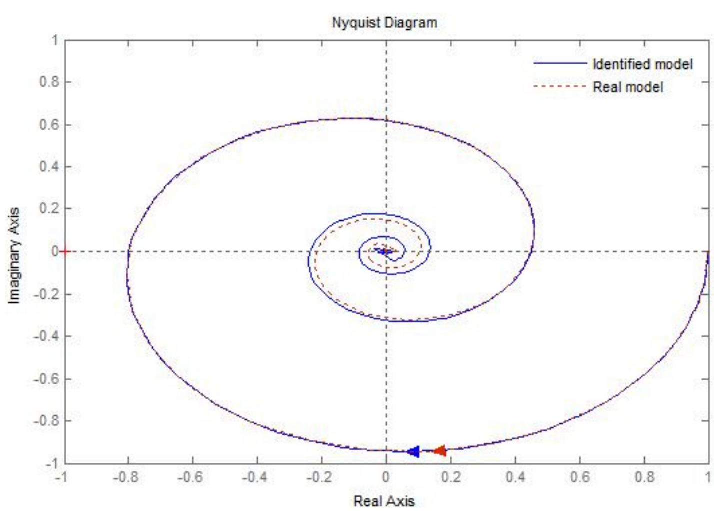

Figure-11
Third order plus dead time system

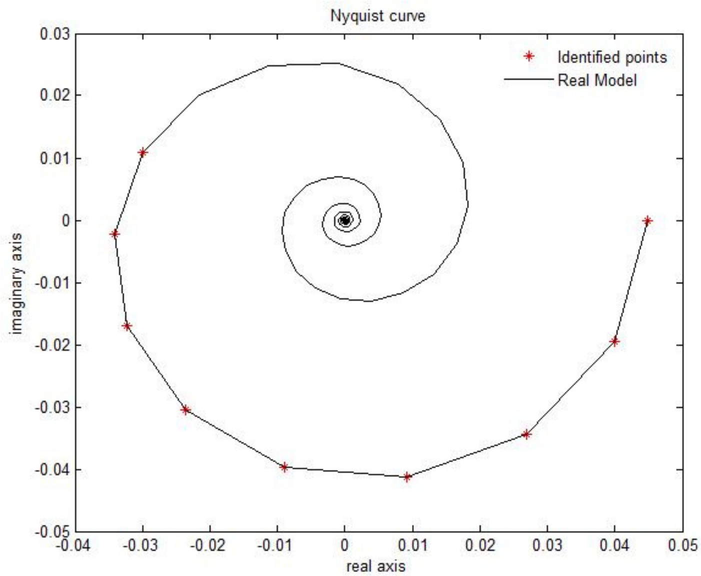

Figure-12
Frequency response of model with long dead-time

In this example, the number of generations and Population size used for genetic algorithm are: 500 and 100 respectably.

Conclusion

The genetic algorithm was an excellent tool to solve the optimization problem. I was very important that same methodology can be used for both cases: PID tuning and transfer function modeling. In both applications, the results obtained were more accurate from the identified points of $G(jw_i+\alpha)$ to $G(jw_i)$; It was due to the fact that using $G(jw_i+\alpha)$ is more direct than $G(jw_i)$. Nonlinear least squares method was successfully applied in all cases to adjust the parameters values in order to reduce the sum of the squares of the errors between the given structure and the measured data points. It is remarkable to say that presented method has a good performance to identify all proposed models: Very long dead time process and both monotonic and oscilatory processes, no matter those different structures were proposed.

It is also important to mention that M_s value was always a referent in relation to a good performance of the designed PID's, especially at the relative stability; on the other hand, when the *Ms* Value is within the proposed range, this ensures that the controlled systems are insensitive to possible changes in plant models[16][3]. So it, the values of Gain Margin and Phase Margin were very close as expected.

References

1. Åström K.J. and Hägglund T. (1995). *PID Controllers:* Theory, Design and Tuning, Second Edition, Instrument Society of America, Research Triangle Park. NC.
2. Kristiansson B. and Lennartson B. (2006). Robust tuning of PI and PID Controllers, *IEEE Control Systems Magazine* 26(1): 55-69.
3. Ziegler G. and Nichols N.B. (1942). Optimum Settings for Automatic Controllers, *Trans. ASME*, 64(11), 759-768.
4. Hang C.C., Åström K.J. and Wang Q.G. (2002). Relay feedback auto-tuning of process controllers: A tutorial review, IFAC, *Journal of Process Control*, 12(1), 143–162.
5. Liu T., Wang Q.G. and Huang H.P. (2013). A tutorial review on process identification from step or relay feedback test, *Journal of Process Control*, 23(1), 1597–1623.
6. Padhy P.K. and Majhi S. (2006). Relay based PI_PID design for stable and unstable FOPDT processes, *Computer and Chemical Engendering*, 30(5), 790-796.
7. Wang Q.C. and Zhang Y. (2001). Robust identification of continuous systems with dead time from step responses, Automatica, 37(1), 377-390.
8. Shin G.W., Song Y.J., Lee T.B. and Choi H.K. (2007). Genetic Algorithm for Identification of Time Delay Systems from Step Responses, *International Journal of Control, Automation, and Systems*, 5(1), 79-85.
9. Liu T. and Shao C. (2012). Closed-loop step identification of low-order continuous-time process model with time delay for enhanced controller autotuning, *Int. J. Systems, Control and Communications*, 4(4), 225-249.
10. Liu T. and Gao F. (2010). A frequency domain step identification method for continuous-time, *Journal of Process Control*, 20(1), 800-809.

11. Gavin H.P. (2013)., The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems, *Department of Civil and Environmental Engineering Duke University*.

12. Griva S.G. Nash and Ariela S. (2009). Linear and Nonlinear Optimization, Second Edition. Society for Industrial Mathematics, United States of America.

13. Åström K.J. and Hägglund T. (1984)., Automatic Tuning of simple controllers with specification on phase and amplitude margins, Automatica, 20(5), 645-51.

14. Jamshidi M., Coelho L. (2003). Santos Dos, Krohling R.A. and Fleming P.J., Robust Control Systems with Genetic Algorithms, CRC Press LLC.

15. Wang Q.G., Chieh CC and Bi Q. (1997)., A Frequency domain controller design method, *Chemical Engineering Research and Design,* 75, 64-72.

16. Coello Coello C.A., Van Veldhuizen D.A. and Lamont G.B. (2007). Evolutionary Algorithms for Solving Multi-Objective Problems, Springer.

17. Fleming P.J. and Purshouse R.C. (2001). Genetic Algorithms in Control Systems Engineering, Department of Automatic Control Systems Engineering University of Sheffield, SI 3JD, Research Report No. 789.

18. Goldberg D.E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley.

19. Wang Q.G. (1997)., Process Frequency Response Estimation from Relay Feedback, *Control Eng. Practice*, 5(9), 1293-1302.

20. Zhou K. and Doyle J.C. (1998). Essentials of Robust Control. Prentice Hall.

21. Doyle J., Francis B.A. and Tannenbaum A.R. (1992). Feedback Control Theory, Macmillan Publishing Company.