CLOSED LOOP SYSTEM IDENTIFICATION USING GENETIC ALGORITHM

Lucchesi Alejandro[(a)], Campomar Guillermo[(b)], Zanini Aníbal[(c)]

[(a,b)] Facultad Regional San Nicolás – Universidad Tecnológica Nacional (FRSN-UTN), Argentina
[(c)] Universidad de Buenos Aires - CONICET, Argentina

[(a)]alucchesi@frsn.utn.edu.ar, [(b)]gcampomar@frsn.utn.edu.ar, [(c)]azanini@fi.uba.ar

ABSTRACT

Time-delay processes are frequently found in industry and the most common representation is a first order plus delay time (FOPDT) transfer function. The identification of time delay systems is a challenging task. Usually, the process has to be disturbed in order to excite enough the system to provide the information for identification. This work presents a genetic algorithm to identify the system using closed loop information. Therefore, a setpoint change in the reference is used, avoiding disturbances to the process.

Keywords: time-delay system, identification, genetic algorithm

1. INTRODUCTION

Despite of the growth of Model Predictive Controllers (MPC) in the industry, the PID (Proportional-Integral-Derivative) is the dominant feedback control algorithm. In most MPC applications, the manipulated variables are the setpoints of PID controllers. Therefore, the performance of the regulatory control loops is important. As indicated by Desborough and Miller (2002), only one third of the controllers performance found in the continuous process industry are acceptable. Lots of benefits can be obtained by improving performance of PID loops, and the identification of the systems dynamics is the first step.

Yang and Seested (2013) used a genetic algorithm method for identifying first order plus dead time (FOPDT) transfer functions and compared the results with other identification techniques using open loop data and different excitation signals.

The present work consist in the study and implementation of a real coded genetic algorithm to identify FOPDT transfer functions from the input-output data obtained from a closed loop step test. Therefore, correlated information will be used to identify the process, as data is obtained from a setpoint change in the controller. Considering a system which is controlled by a PID algorithm, three vectors containing information of sample time, controller output and system output are necessary to feed the identification software.

The structure of this paper is as follows. In this section the mathematical model of the system is presented. In Section 2, a description of the genetic algorithm for the identification. In Section 3, the data set generation for testing the algorithm. In section 4, the results and a test using data from a closed loop system in an industrial facility and conclusions in Section 5. Finally, future works in Section 6.

1.1. Plant model

A first order plus dead time model (FOPDT) is used (1).

$$G_{(s)} = \frac{K.e^{-\theta s}}{\tau s + 1} \tag{1}$$

The discrete form of the model preceded by a ZOH is described in (2).

$$G_{(z)} = \frac{b}{z - a}.z^{-d} \tag{2}$$

$$\hat{y}_k = a.\hat{y}_{k-1} + b.u_{k-1-d} \tag{3}$$

where

$$b = K.\left(1 - e^{-\frac{Ts}{\tau}}\right) \qquad a = e^{-\frac{Ts}{\tau}} \qquad d = \left\lfloor \frac{\theta}{Ts} \right\rfloor \tag{4}$$

2. THE GENETIC ALGORITHM FOR IDENTIFICATION

In the FOPDT model, the parameters to be identified are gain, delay time and time constant. Yang and Seested (2013) showed that the identification of unknown time delay systems is often a non-convex optimization problem and that the cost surface could have more than one minimum. They also showed that better results are obtained while using the continuous time parameters rather than the discrete ones, because of the exponential relationship between them (4) that affects the performance of the genetic algorithm.

In the algorithm, the output of each system in the population is calculated using the real data input vector.

The cost function to minimize is defined as the sum of the absolute error between the real output and the model estimated output.

$$J = \sum_{k=1}^{N} \left| y_k - \hat{y}_k \right| \qquad (5)$$

2.1. The genetic algorithm

The genetic algorithm is a well known tool to solve optimization problems. It's based in principles of genetics and natural selection. It's composed by a population of individuals, that evolves on each iteration to minimize (or maximize) a cost function.

One of the most important benefits of the genetic algorithm is that it can handle a complex cost surface and its possibilities to jump out of a local minimum.

The algorithm was introduced by Holland (1975) and De Jong (1975) and popularized by Goldberg (1989). The population of solutions can be described as in (6), and the best solution at the time t is x_i^t.

$$P_{(t)} = \left\{ \vec{x}_1^{\,t}, \ldots\ldots, \vec{x}_N^{\,t} \right\} \qquad (6)$$

Where

$$\vec{x} = \left\{ K, \tau, \theta \right\} \qquad (7)$$

The initial values of P are randomly generated between the established limits for each variable.

At each iteration, the population evolves through the application of the genetic operators: heuristic crossover, static mutation and dynamic mutation.

Janikow and Michalewicz (1991) described the benefits regarding processing speed and precision of the real coded algorithm compared to the binary coded one. Therefore, a real coded genetic algorithm is used. The population number is set to 50, and the execution is limited to 200 iterations. The survival rate is 60% and the mutation rate is set to 20%.

2.1.1. Heuristic crossover

On each iteration, the population is sorted by the function cost, and natural selection occurs. The worst 20 individuals are discarded, to make room to the new ones. 20 pairs of individuals are selected randomly to generate the offsprings to be included in the population.

The heuristic crossover method, which is described in Wright (1990), is used. The function is (8)

$$\vec{x}_3 = \beta\left(\vec{x}_2 - \vec{x}_1\right) + \vec{x}_2 \qquad (8)$$

Where β is a random number between 0 and 1, and with $\vec{x}_2$ better than $\vec{x}_1$.

With this function exists the possibility to create an offspring outside the limits. This condition must be checked, and in case it occurs, another offspring should be generated with a new random number.

With this operator, the information of the better solutions is combined to generate new solutions. It explores a point by moving outward of the better parent. The constrain is that no new information is introduced in the population.

2.1.2. Static mutation

The genetic algorithm can converge to a local minimum solution quickly. To avoid this issue, the mutation operator is introduced, which enables the exploration of other surfaces of the cost function by producing random changes in the variables. As the mutation rate is set in 20%, 10 variables will be changed by random numbers on each iteration. This operator is important in the early phases of the algorithm, as searches on different surfaces areas.

2.1.3. Dynamic mutation

When the iteration number is in 100, the dynamical mutation described by Janikow and Michalewicz (1990) is introduced. The operator is described in (9).

$$x'_k = \begin{cases} x_k + \Delta\left(t, UB - x_k\right) & \textit{if a random bit is } 0 \\ x_k - \Delta\left(t, x_k - LB\right) & \textit{if a random bit is } 1 \end{cases} \qquad (9)$$

Where LB and UB are the limits of the variable. The function $\Delta(t, x)$ is defined in (10)

$$\Delta(t, x) = x.r.f\left(1 - \frac{t}{T}\right) \qquad (10)$$

Where f is a scale factor equal to 0.01, r is a random number between 0 and 1, and T is the maximal iteration number.

This operator improves the "fine tuning", as the searched space decreases at the later stages of the population life.

2.1.4. Genetic algorithm flowchart

In Figure 1 a flow chart describing the genetic algorithm implementation is shown.

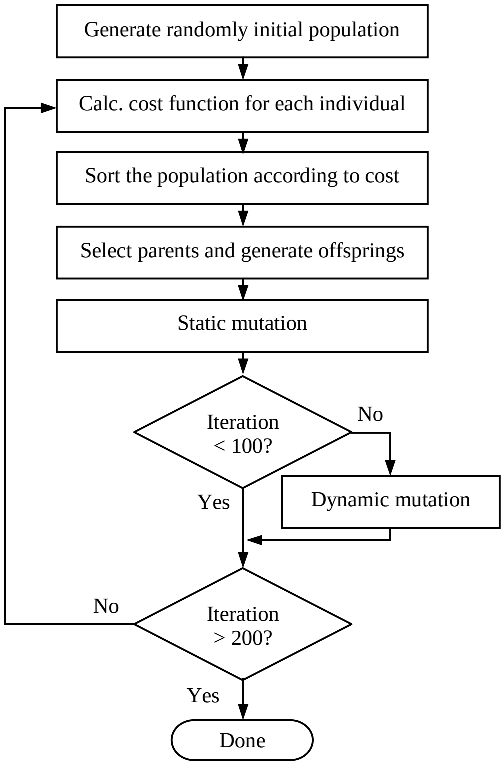

Figure 1: genetic algorithm flowchart.

2.2. Variables limits

The boundaries of the searched variables are defined. For the gain, the maximum and minimum values of the input and output vectors are used, with a margin of 10, as indicated in (11)

$$k_{LIM} = \pm \frac{\max\{y\} - \min\{y\}}{\max\{u\} - \min\{u\}} \times 10 \qquad (11)$$

The delay time limit is calculated as (12)

$$\theta_{LIM} = \frac{N}{Ts} \qquad (12)$$

With N the length of the vector.
The constant time limit is calculated as (13)

$$\tau_{LIM} = 5 \times \theta_{LIM} \qquad (13)$$

3. TESTING DATA

3.1. Generation of the testing data

For the testing of the identification algorithm, three different transfer functions are used, as described in Table 1, with a relationship between the time delay and time contant of 0.2, 1 and 5.

Table 1: parameters of the testing transfer functions

	System 1	System 2	System 3
Gain (k)	0.8	0.8	0.8
Delay (θ)	4	10	20
Time constant (τ)	20	10	4
Delay (θ) / Time constant (τ)	0.2	1	5

Three different PI controllers are designed, in order to obtain underdamped, critical damping, and overdamped simulated closed loop information.
Three different levels of white noise are added to the process variable, with a variance of 0 (no noise), 0.001 and 0.01.
Therefore, 27 sets of simulated closed loop information are used to test the identification algorithm.

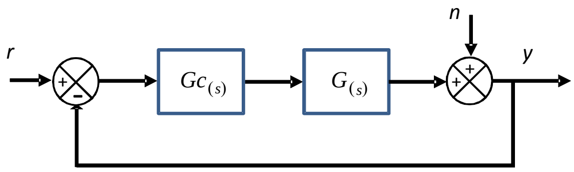

Figure 2: simulation block diagram for testing data generation.

4. RESULTS

4.1. Gain parameter

The identification of the gain parameter can be seen in Figure 3. The value is normalized to 1.

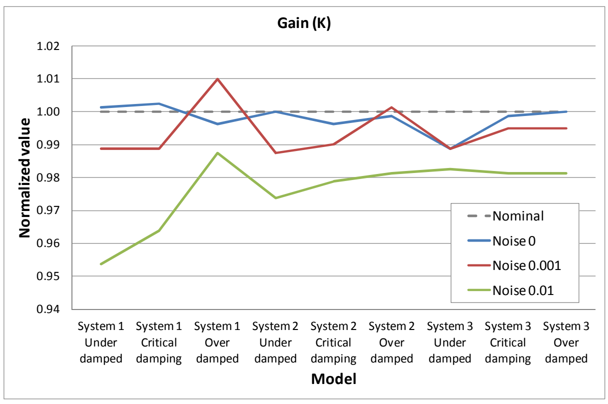

Figure 3: gain parameter for different models and noises variance.

4.2. Time delay parameter

The identification of the time delay parameter can be seen in Figure 4. The value is normalized to 1.

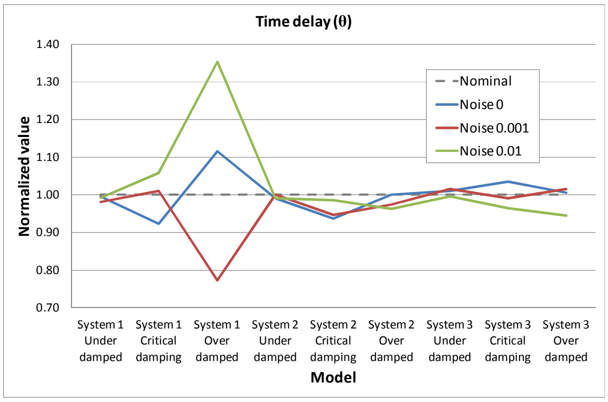

Figure 4: time delay parameter for different models and noises variance.

4.3. Time constant parameter

The identification of the time constant parameter can be seen in Figure 5. The value is normalized to 1.

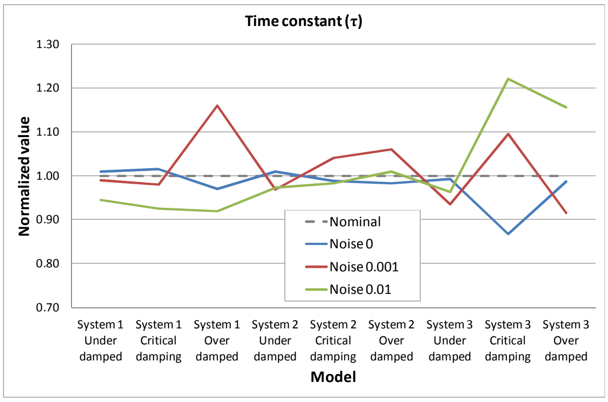

Figure 5: time delay parameter for different models and noises variance.

4.4. Noise influence

The influence of noise in the identification is shown in figure 6. It can be seen that for higher noise, the standard deviation of the estimated parameters is higher, showing the negative impact of noise.

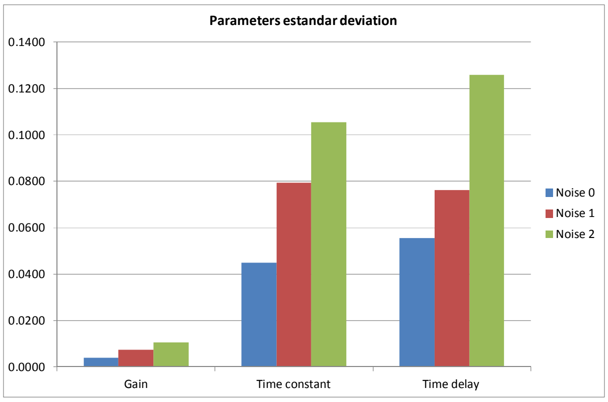

Figure 6: standard deviation of the estimated parameters under different noises variances.

4.5. Influence of the controller tuning parameters of the closed loop data

The controller "a" is the one tuned with the more aggressive parameters, with an underdamped closed loop response. The identifications with the information from this controller show the minimum standard deviation in the time constant and time delay parameters figure 7.

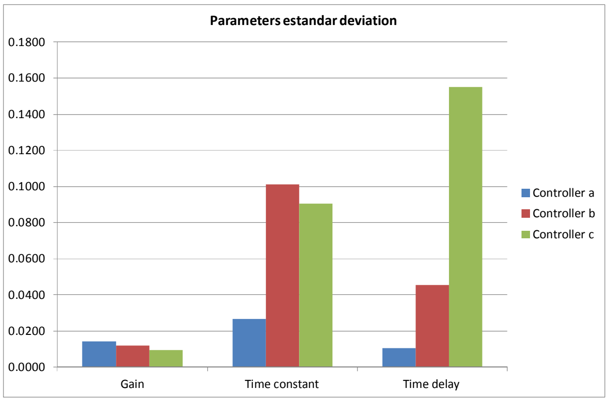

Figure 7: standard deviation of the estimated parameters under different controllers.

4.6. Examples of responses for the higher noises

The identification results are shown as an example for data with higher noise. It can be seen the data from the simulation used for the identification, and the identified system output.

In figure 8 the system with $K = 0.8$ / $\tau = 20$ / $\theta = 4$ is used, with a controller generating an underdamped response.

In figure 9 the system with $K = 0.8$ / $\tau = 10$ / $\theta = 10$ is used, with a controller generating a critical damping response.

In figure 10 the system with $K = 0.8$ / $\tau = 4$ / $\theta = 20$ is used, with a controller generating an overdamped response.

It can be seen the data from the simulation used for the identification, and the identified system output.

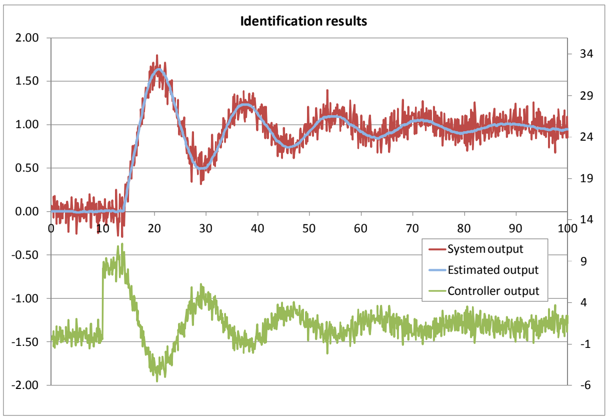

Figure 8: identification results for model 1.

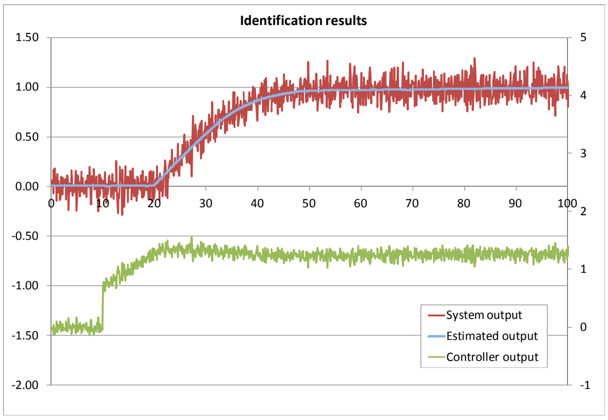

Figure 9: identification results for model 2.

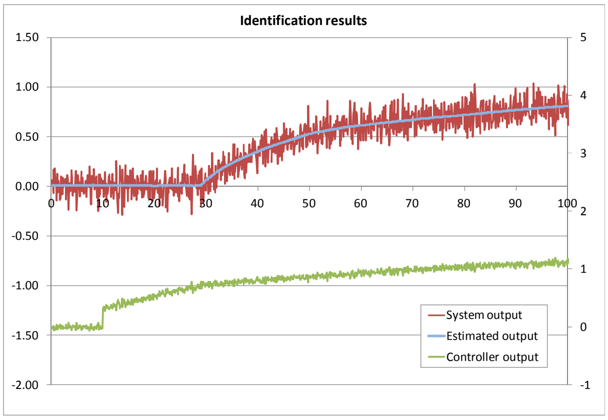

Figure 10: identification results for model 3.

4.7. Test of the algorithm with industrial data.
A closed loop information from an industrial facility is used to test the algorithm. The production site is an air separation unit (ASU) that produces gaseous oxygen by cryogenic distillation. The purity is controlled with the oxygen flow production. Even though the loops are in cascade (the purity controller Gcp output is the setpoint of the flow controller Gcf), the dynamics of the flow loop are much faster than the purity one. Therefore, the data from the output of the master purity controller is used as input vector (setpoint of the flow controller), and the purity as output vector.

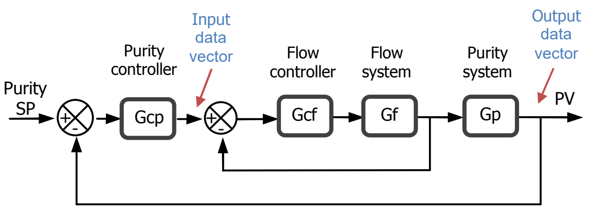

Figure 10: data collection block diagram.

A purity setpoint step from 97.3% to 97.2% is used for identification. The results are:

K = -0.0056

τ = 130.29 min

θ = 51.24 min

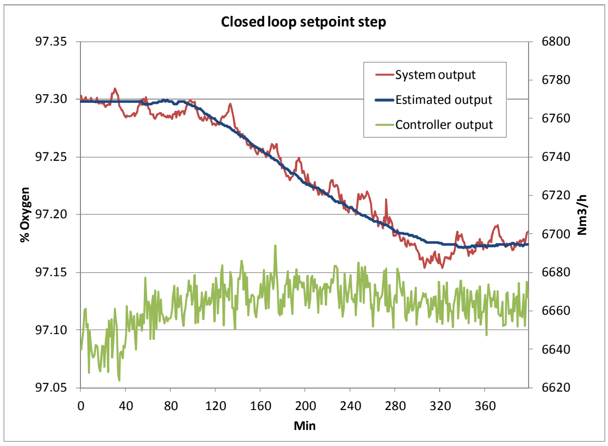

Figure 11: identified system from industrial facility.

The identification software had to be adapted for the real data usage. The Setpoint vector was included, in order to detect a change in the reference. The system data, until the setpoint change, is considered in steady state, and the average of those samples are calculated as in (14) and (15). Then, those values are substracted from the u and v vector.

$$u_0 = \frac{\sum_1^k u_k}{k} \tag{14}$$

$$y_0 = \frac{\sum_1^k y_k}{k} \tag{15}$$

This kind of loops, where the delay time and time constant values are relatively high, are difficult to tune with a trial and error approach, and it's common to see them in Manual mode in the industries.

5. CONCLUSIONS

The algorithm showed a good performance and robustness. Even with the higher noise variance, the estimated parameters are acceptable and suitable for controller design. The results obtained on a real system were satisfactory and allow the identification application for closed-loop tuning controllers.

6. FUTURE WORK

After showing the capabilities of the genetic algorithm in the identification of FOPDT systems, the objective is to extend the scope to integrating process with time delay and second order plus delay time (SOPDT).
Several rules exist for controller design based on the transfer function parameters, as described by O'Dwyer (2000a and 2000b). In the future, the design of the controller using optimization methods based on a desired closed loop response will be proposed.

REFERENCES

De Jong, K. A. 1975. Analysis of the behavior of a class of genetic adaptive systems. Ph.D. Dissertation. University of Michigan, Ann Arbor.

Desborough L., Miller R., 2002. Increasing Customer Value of Industrial Control Performance Monitoring—Honeywell's Experience. AIChE Symposium Series. pp. 169–189.

Goldberg,D. E. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley.

Holland, J. H. 1975. Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan Press.

Janikow C.Z., Michalewicz, Z., 1991. An experimental comparison of binary and floating point representations in genetic algorithms. Proceedings of the 4th International Conference on Genetic Algorithms, San Diego, CA, USA.

Janikow C.Z., Michalewicz Z., 1990. A specialized genetic algorithm for numerical optimization problems," Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence, p.798-804.

O'Dwyer A., 2000a. A summary of PI and PID controller tuning rules for processes with time delay. Part 1: PI controller tuning rules. Proceedings of PID '00 IFAC Workshop on Digital Control, pp. 175-180, Terrassa, Spain.

O'Dwyer A., 2000b. A summary of PI and PID controller tuning rules for processes with time delay. Part 2: PID controller tuning rules. Proceedings of PID '00 IFAC Workshop on Digital Control, pp. 242-247, Terrassa, Spain.

Wright, A.H., 1990. Genetic Algorithms for Real Parameter Optimization. In Rawlins, G. (Ed.), Foundations of Genetic Algorithms, First Workshop on the Foundations of Genetic Algorithms and Classier Systems, Morgan Kaufmann Publishers, Los Altos, CA, pp.205-218.

Yang Z., Seested G.T., 2013. Time-Delay System Identification Using Genetic Part One: Precise FOPDT Model Estimation. Proceedings of the 3rd IFAC International Conference on Intelligent Control and Automation Science. Vol. 3, pp. 561-567.

AUTHORS BIOGRAPHY

Lucchesi, Alejandro:
Electronic Engineer graduated from the National Technological University College St. Nicholas (UTN-FRSN), Argentina in 2006.
Actually working on his Master degree thesis in numerical simulation and control, from the College of Engineering of the University of Buenos Aires (UBA), Argentina.
Assistant professor of control systems in Electronic Engineering career and researcher of the control systems study group (GESiC) in the UTN - FRSN since 2002.

Campomar, Guillermo:
Electronic Engineer graduated from the National Technological University College St. Nicholas (UTN-FRSN), Argentina in 2001 .
Magister in numerical simulation and control, graduated from the College of Engineering of the University of Buenos Aires (UBA), Argentina in 2006 .
Professor of control systems in Electronic Engineering career and coordinator of the control systems study group (GESiC) in UTN - FRSN since 2002.

Zanini, Aníbal:
Electrical Engineer from National University of Rosario – Argentina (1977) and the Phd. in Automatica from Universidad Politécnica de Madrid – Spain (1983).
Associate Professor with University of Buenos Aires – Argentina.