15th Triennial World Congress, Barcelona, Spain

OPTIMAL PID TUNING WITH GENETIC ALGORITHMS FOR NON LINEAR PROCESS MODELS

J.M. Herrero, X. Blasco, M. Martínez, J.V. Salcedo [1]

Predictive Control and Heuristic Optimization Group
Department of Systems Engineering and Control
Universidad Politécnica de Valencia
Camino de Vera 14, P.O. Box 22012 E-46071 Valencia, Spain
Tel: +34-963877000 ext: 5713 . Fax: +34-963879579.
E-mail: xblasco@isa.upv.es http://ctl-predictivo.upv.es

Abstract: This work presents a powerful and flexible alternative for tuning PID controllers using Genetic Algorithms. The potential of this technique is shown using non-linear process models and a reference trajectory. Flexibility is demonstrated by showing how to tune an optimal PID in various situations: model errors, noisy input, IAE minimization, and following a reference models, etc. These problems are solved by changing the minimization index. *Copyright ©2002 IFAC.*

Keywords: Genetic Algorithms, Optimization, PID, Robust Control, Non-linear Control.

1. INTRODUCTION

PID controllers are the most common controllers in industry, in fact, 95% of control loops use PID and the majority are PI. (Aström and Hägglund, 1995). Accordingly, there are many tuning techniques, and most are based on:

(1) Empirical methods, such as Ziegler-Nichols methods. (Aström and Hägglund, 1995).
(2) Analytical methods, for instance, the root locus based techniques. (Aström and Hägglund, 1995) (Blasco *et al.*, 2000).
(3) Methods based on optimization, such as Ciancone or Lopez methods (Marlin, 1995). These obtain PID parameters by optimising an IAE index and a linear model with the following structure.

$$G(s) = \frac{k_p}{\tau s + 1} e^{-\theta s}$$

In all of these cases, PID tunings are obtained for an operation point where the model can be considered linear. This implies there is sub-optimal tuning when a process operates outside the validity zone of the model. This situation is common when the reference is not a set point but a trajectory (robot control, heating trajectories in furnaces, etc.).

An alternative method to solve this problem is to obtain a model for different operational zones, tune a PID controller for each, and establish a mechanism for changing from one controller to another depending on the operation zone (gain planning).

Another alternative is tuning a PID controller by taking into account all non-linearities and additional process characteristics. At this point appears the idea of using Genetic Algorithms (Herreros *et al.*, 2000) (a global optimization technique) to obtain a PID tuning that meets all the requirements established in a minimization index by the designer.

[1] This work has been partially financed by European FEDER funds, project 1FD97-0974-C02-02.

2. GENETIC ALGORITHMS

Genetic Algorithms (GA) (Goldberg, 1989), (Holland, 1975) are optimization techniques based on simulating the phenomena that takes place in the evolution of species and adapting it to an optimization problem. These techniques imply applying the laws of natural selection onto the population to achieve individuals that are better adjusted to their environment.

The population is nothing more than a set of points in the search space. Each individual of the population represents a point in that space by means of his chromosome. The individual's degree of adaptation is given by the objective function.

Applying genetic operators to an initial population simulates the evolution mechanism of individuals. The most usual operators are as follows:

- Selection: The main goal is selecting the chromosomes with the best qualities for integration in the next generation (these would depend on the cost function for each individual).
- Crossover: By combining the chromosomes of two individuals, new chromosomes are generated and integrated into the population.
- Mutation: Random variations of parts of the chromosome of an individual in the population generate new individuals.

The variations of the Genetic Algorithms can be distinguished by the kind of codification used for the chromosomes and the genetic operators used.

GA have demonstrated very good performances as global optimisers in many types of applications (Michalewicz, 1996), (Blasco *et al.*, 1998) (Blasco, 1999).

3. OPTIMAL PID FOR A THERMAL PROCESS CONTROL

A thermal process (figure 1) is used to illustrate the application of Genetic Algorithms to PID controller tuning. This process presents non-linearities, saturation model errors, and input noises. The state space representation is used and the parameters are obtained by identification using GA (**?**):

$$\dot{x_1} = k_1 \cdot u^2 - k_2(x_1 - T_{in})$$
$$\dot{x_2} = \frac{1}{16.75}(x_1 - x_2)$$
$$\hat{y} = x_2$$

where:

- $\hat{y}$: resistance temperature in ^oC (Controlled variable).
- u: voltage percentage at the control of the actuator (manipulated variable).
- T_{in}: air temperature (in ^oC) in the resistance neighbourhood (Disturbance).
- x_1, x_2: temperatures (in ^oC) involved in heating exchange (internal variables).
- k_1, k_2: model parameters. Nominal parameters are $k_1 = 86 \cdot 10^{-6}$ and $k_2 = 6.027 \cdot 10^{-3}$.

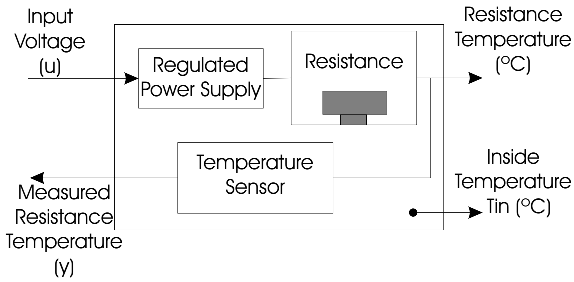

Fig. 1. Thermal Process Scheme.

As usual, control tuning has to achieve different types of specification, such as:

(1) Obtaining dynamic performances evaluated with:
   - minimization of a performances index such as IAE (integral of absolute value error).
   - Adjusting time specifications.

(2) Obtaining robustness properties:
   - Model error robustness.
   - Input noise robustness.

GA is simply a global and powerful optimization technique and, to use it in the tuning process, it is necessary translate all these types of specification to a cost function that the GA has to minimize. The advantage of a GA is that it is possible to solve very complicated cost functions and that allows the PID designer to establish any type of specification.

The following explains how to adjust the PI controller (two parameters, K_c and T_i)[2] for different types of specification in order to control the thermal process.

At the end of this section a summary table with the PI controllers, which has been designed for the IAE minimization, model error and input noise robustness, will be shown.

3.1 *PI tuning for reference model*

In this case, the objective is to obtain a controlled variable following a time specification[3]. This type of specification can be represented as an ideal closed loop model - called the reference model.

Now the tuning process has to minimize, over a simulation time (t_{sim}), differences between reference model output ($y_{mr}(t)$) and the non-linear model controlled by the PI output ($y(t)$), see figure 2. This is represented as a cost function:

---

[2] PI used is an ISA standard structure with anti-windup. $PI(s) = k_c(1+1/(T_i s))$

[3] Restricción on: settling time t_e, overshoot δ, steady-state errors, etc.

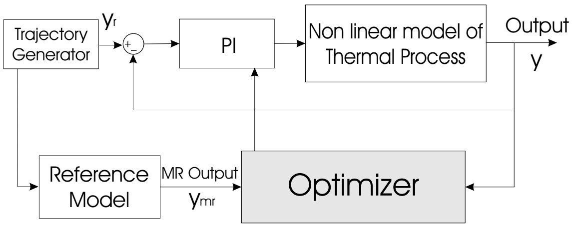

Fig. 2. Tuning structure for reference model.

$$J(K_c, T_i) = \sum_{t=0}^{t_{sim}} |y_{mr}(t) - y(t)| \quad (1)$$

Reference models used in this examples, have been selected to achieve these specifications:

- Settling time $t_e(98\%) = 100 seconds$
- Overshoot $\delta = 0\%$
- Steady-state error $e_p = 0$

$$MR(s) = \frac{1}{(25s+1)} \frac{C}{\%} \quad (2)$$

The optimization problem is two-dimensional (two parameter K_c, T_i), there is no guarantee that it is a convex problem (non-linear model, saturation, PI anti-windup, etc.), and there is no computational restriction (off line optimization). Therefore, GAs offer a good alternative for solving the problem.

GA characteristics are:

- Real codification chromosome.
- Number of individual: 400.
- Ranking operator.
- Selection operator: Stochastic universal sampling.
- Crossover operator: Intermediate recombination with a probability of 0.85.
- Mutation operator: Random with a Gauss distribution ($\sigma = 2\%$) and a probability of 0.05.
- Search space: $K_c \in [0.1 \ldots 20]$ $\frac{^oC}{\%}$ and $T_i \in [10 \ldots 600] sec$.

The solution is:

$$K_c = 2.14 \; , \; T_i = 559.97 \; , \; J(K_c, T_i) = 66.24$$

Figure 3 shows K_c, T_i and $J(K_c, T_i)$ evolution during the optimization process. It shows optimization process convergence in the 20th iteration. The high number of individuals and the GA operator selected assure a good search space exploration, and that means the solution has a high degree of confidence.

Figure 4 shows control results obtained with this solution and compared with the reference model.

3.2 *PI tuning for IAE minimization*

The objective is adjusting PI parameters K_c and T_i to minimize the IAE index of closed loop response (figure 5). That means, minimizing the following cost index:

$$J(K_c, T_i) = \sum_{t=0}^{t_{sim}} |y_r(t) - y(t)| \quad (3)$$

Where $y_r(t)$ is the reference trajectory, $y(t)$ is the model closed loop simulation with PI controller, and t_{sim} is the simulation time.

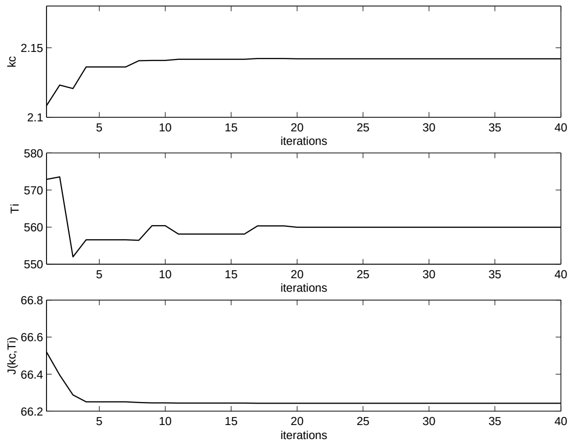

Fig. 3. K_c, T_i and $J(K_c, T_i)$ evolution for reference model optimization.

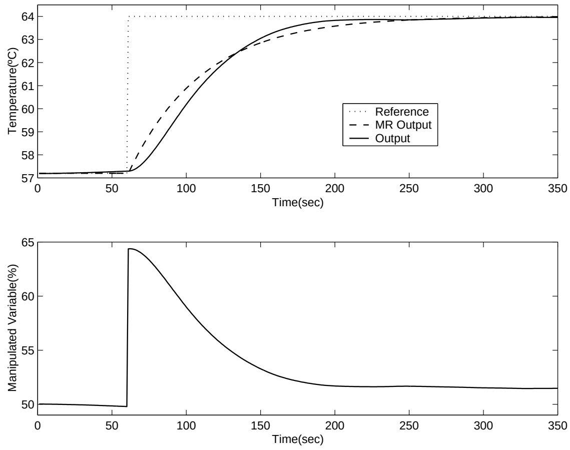

Fig. 4. PI closed loop control for reference model.

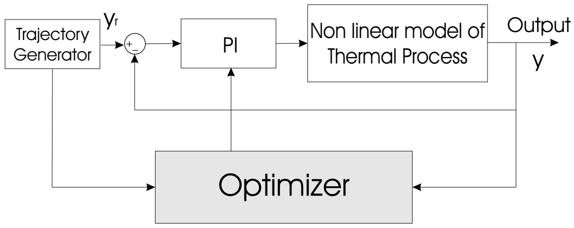

Fig. 5. Tuning structure for IAE minimization.

The GA characteristics are the same as in the previous problem and the solution is:

$$K_c = 9.18 \; , \; T_i = 179.13 \; , \; J(K_c, T_i) = 533.07$$

Figure 7 shows optimization convergence behaviour is good. Figure 6 shows control result with the PI obtained.

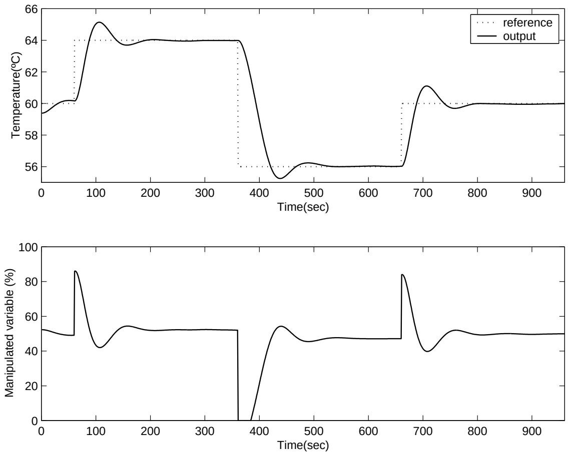

Fig. 6. PI closed loop control for IAE minimization.

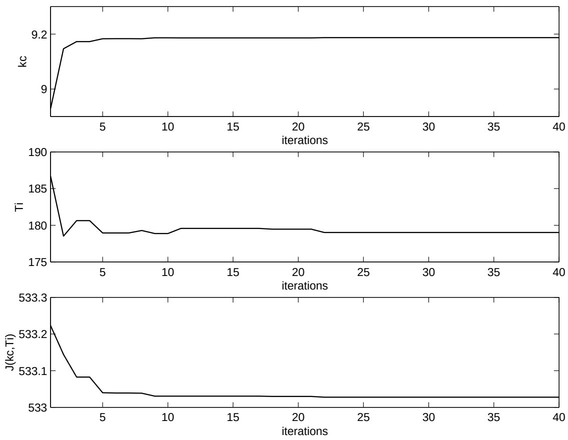

Fig. 7. K_c, T_i and $J(K_c,T_i)$ evolution for IAE minimization.

3.3 *PI tuning for model error robustness*

A method to increase robustness when there is model error, is by introducing in the cost index the minimization of the nominal process behaviour (IAE or reference model), $y_1(t)$, and introducing the behaviour of several variations of the nominal process, $y_j(t)\ j \neq 1$, (changing model parameters value). In this way, the cost index for IAE minimization follows a reference trajectory ($y_r(t)$) and increasing robustness against model errors could be:

$$J(K_c,T_i) = \sum_{j=1}^{n} \sum_{t=0}^{t_{sim}} |y_r(t) - y_j(t)| \quad (4)$$

For instance, the PI of the thermal process is tuned while assuming three situations: firstly, ($j = 1$) nominal parameters value, secondly, ($j = 2$) an increase of 30% from nominal value in parameter k_1 and thirdly, ($j = 3$) an increase of 25% in k_2. Obviously, other situations can be added if necessary.

j	k_1	k_2
1	0.000086	0.006027
2	0.000110	0.006027
3	0.000086	0.0075

PI parameters obtained are:

$$K_c = 10.68\ ,\ T_i = 136.84\ ,\ J(K_c,T_i) = 1645.90$$

Figure 8 shows closed loop control results with the optimal PI obtained. The IAE obtained for the nominal process is 540.14 which is greater than the 533.07 obtained when it was only optimised for the nominal process. Figure 9 shows evolution during the optimization process.

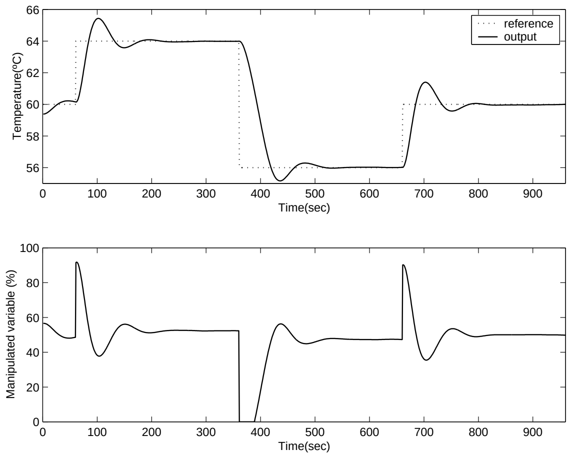

Fig. 8. PI closed loop control for IAE minimization with model error robustness.

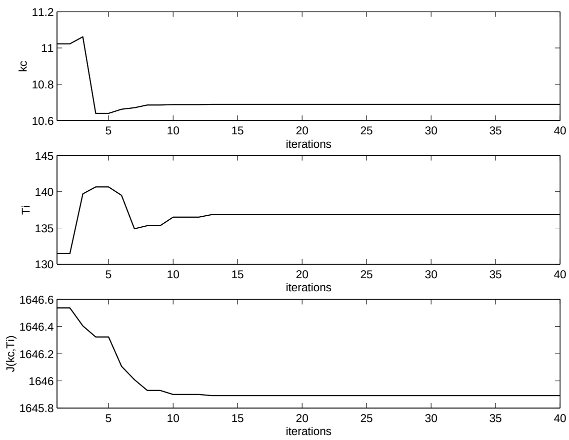

Fig. 9. K_c, T_i and $J(K_c,T_i)$ evolution for IAE minimization with model error robustness.

If model errors are present and the process parameters are:

$$k_1 = 0.000070\ ;\ k_2 = 0.0100$$

figure 10 compares nominal and robust PI closed loop simulations. Now the IAE obtained for the robust PI is 709.79, but the nominal PI IAE case obtained 866.49.

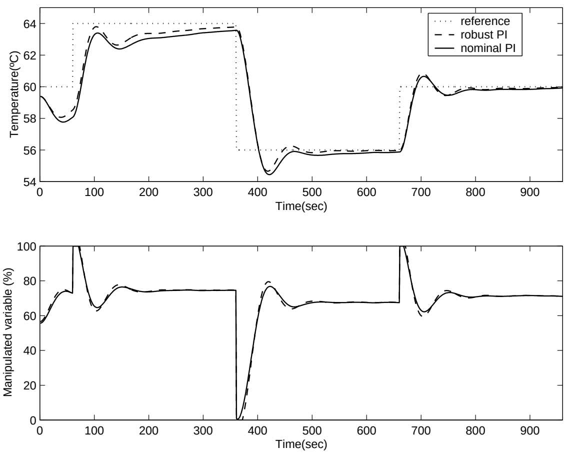

Fig. 10. Closed loop control with nominal and robust PI.

3.4 *PI tuning for input noise*

Another important source of problems in processes can be input noise. It can be possible to take this into account if the tuning optimization process simulates a noise of similar characteristics to that observed in the process.

Cost index shows no changes in the IAE minimization and reference model after changing nominal behaviour ($y(t)$) for noisy behaviour ($y_n(t)$), simply adding a noise ($n(t)$) with the same characteristics as the real one:

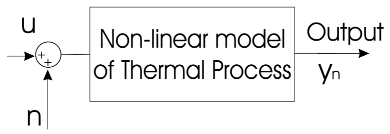

Fig. 11. Structure with input noise.

The cost index for a IAE minimization is:

$$J(K_c, T_i) = \sum_{t=0}^{tsimul} |y_r(t) - y_n(t)| \qquad (5)$$

For thermal processes, the noise is a random signal with a normal distribution of 10% amplitude.

The obtained result is:

$$K_c = 11.34 \ , \ T_i = 281.70 \ , \ J(K_c, T_i) = 544.79$$

Figure 12 shows closed loop control results with the tuned PI. The IAE obtained for the nominal process without input noise is 542.45 which is greater than the 533.07 obtained when it was optimised only for nominal process. Figure 13 shows K_c, T_i and $J(K_c, T_i)$ evolution in the optimization process.

With a normal distribution noise of 10% amplitude, the simulations are repeated for the nominal and noise PI. Obtained results are shown in figure 14. Now the IAE obtained for the noise PI is 544.79, but in the nominal PI case the IAE obtained is 600.67.

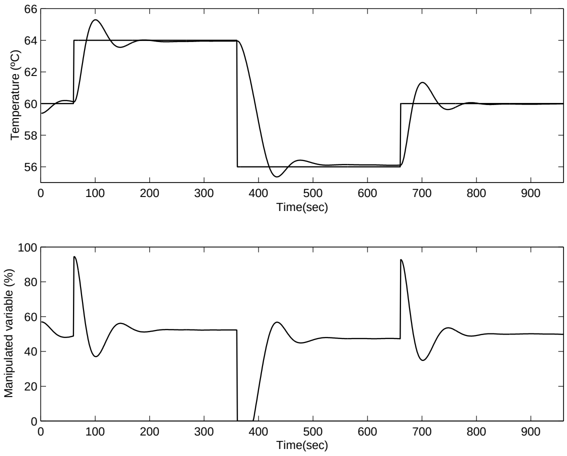

Fig. 12. Closed loop nominal process control with PI for noise robustness.

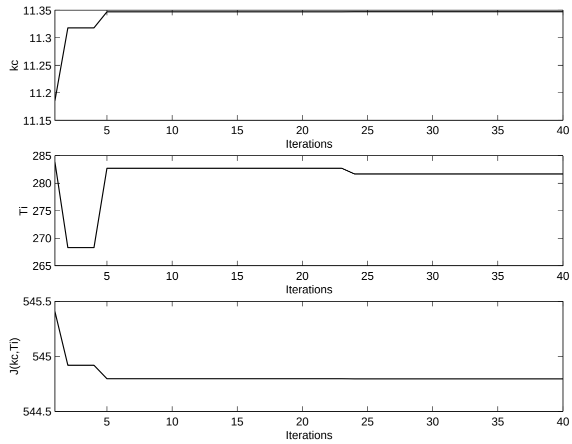

Fig. 13. K_c, T_i and $J(K_c, T_i)$ evolution for IAE minimization with input noise.

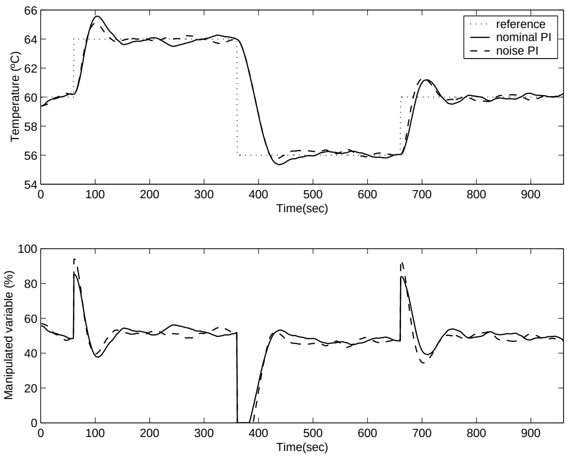

Fig. 14. Closed loop control with nominal and noise PI.

The following table shows the results obtained through the section.

PI Parameters k_c, T_i	Nominal Process $J(k_c, T_i)$	Model Error $J(k_c, T_i)$	Input Noise $J(k_c, T_i)$
9.18,179.13	533.07	866.49	600.67
10.68,136.84	540.14	709.79	...
11.34,281.70	542.45	...	544.79

4. CONCLUSIONS

This work shows how simple and powerful a GA application for controller tuning can be. Because the GA is a very good optimization technique, all control specifications that can be translated to a cost index can be applied.

Application for different performance specifications (IAE minimization and restrictions in time domain) and robustness quality improvement (model errors and input noises) are shown. Everything applies to a non-linear process.

Only the application for a PID industrial controller is shown because it is one of the most important basic controllers. However, this technique can be applied to many linear and non-linear controllers. It is also possible to extend application to a multivariable control by simply adapting the cost index function. The only limitation is the computational cost of the optimization process - however, for off-line tuning this is not a major problem.

5. REFERENCES

Aström, K.J. and T. Hägglund (1995). *PID Controllers: Theory, Design, and Tuning*. Instrument Society of America.

Blasco, F.X. (1999). Model based predictive control using heuristic optimization techniques. Application to non-linear and multivariables proceses (In Spanish). PhD thesis. Universidad Politécnica de Valencia. Valencia.

Blasco, F.X., M. Martínez, J. Senent and J. Sanchis (1998). Generalized predictive control using genetic algorithms (GAGPC). An application to control of non-linear process with model uncertainty. In: *Methodology and tools in Knowledge-based systems* (Springer, Ed.).

Blasco, F.Xavier, M. Martínez, J. Senent and J. Sanchis (2000). *Sistemas Automáticos*. Editorial U.P.V.

Goldberg, D.E. (1989). *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley.

Herreros, A., E. Baeyens and J.R. Perán (2000). Design of PID controllers using multiobjective genetic algorithms. In: *Workshop on Digital control: Past, present and future of PID Control, PID'00* (Preprints, Ed.).

Holland, J.H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press.

Marlin, T. E. (1995). *Process Control. Designing Processes and Control Systems for Dynamic Performance*. Mc Graw-Hill.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer.