Educational Tool for Optimal Controller Tuning Using Evolutionary Strategies

Daniel Carmona, Jorge E. Jiménez-Hornero, Francisco Vázquez and Fernando Morilla

***Abstract*—In this paper, an optimal tuning tool is presented for control structures based on multivariable PID (Proportional–Integral–Derivative) control, using genetic algorithms as an alternative to traditional optimization algorithms. From an educational point of view, this tool provides students with the necessary means to consolidate their knowledge on these control structures, which are of particular relevance in control engineering. The Graphical User Interface designed for the tool allows for: the selection of the control structure, the desired decoupling, the type of PID, the analysis of the interaction effects through RGA (Relative Gain Array), the planning of several optimal tuning processes, the comparison of different designs (through graphics or the numeric results obtained) and the management of data files saved during the planned optimal tunings process. The developed tool was made available to students for them to solve a practical problem and, subsequently, the impact of its use was evaluated.**

***Index terms*—Educational tool, genetic algorithms, multivariable control, optimal tuning, PID**

Mr Carmona, Drs Jiménez-Hornero and Vázquez are with the Departamento de Informática y Análisis Numérico, Universidad de Córdoba, 14071 Córdoba, Spain; Dr Morilla is with the Departamento de Informática y Automática, Universidad Nacional de Educación a Distancia (UNED), 28040 Madrid, Spain

I. INTRODUCTION

MIMO (multiple-input and multiple-output) processes are common in industry. There are various control strategies for these processes, and various ways to tune the controllers that are part of such strategies. Therefore, analysis of MIMO processes is considered a fundamental aspect of control engineer training and, MIMO control systems have been, and are still being, widely investigated. In general, these can be classified into centralized or decentralized control systems. Different decentralized control method techniques have been developed, among them designs based on some SISO methods [1], [2], works that look for critical gains of the system in order to tune the controllers [3]-[7] and methods in which the controllers are obtained by means of analytic, numeric or graphics methods. Centralized control techniques are used when the interaction between variables is so high that the problem cannot be solved with a decentralized control. In this work a set of techniques is used that breaks in two phases the structure of centralized control: first, decoupling the system in order to minimize interaction or to make the system diagonal dominant; then, designing the controllers using a decentralized method [8]. The other technique is based in the use of four PID controllers.

To tune controllers employed in control structures, various techniques can be used, some of which are based on optimization. These optimization-based tuning methods can be classified into two groups: those using traditional optimization algorithms [9]; and those using evolutionary computation algorithms, frequently Genetic Algorithms (GAs). In recent years evolutionary computation algorithms are being used instead of traditional optimization algorithms because they provide a much greater efficiency in obtaining the global optimum and thus minimize the computational cost.

From an educational point of view, it would be useful to have software tools that allow design of control structures from simple, flexible descriptions, and that offer a wide range of options. In this respect, it is noteworthy that various simulation programs already exist to design controllers for MIMO systems using traditional tuning methods [9]. There are also tools that simulate MIMO processes on which is possible to design different control strategies [10], and virtual labs that allow tuning the of a real plant's controllers [11]-[13]. However, it is difficult to find simulation tools for the optimal tuning of controllers for MIMO control systems. It is therefore of interest to develop a tool like that presented here: a software tool developed in Matlab for optimal tuning (through genetic algorithms) of control structures for MIMO systems 2x2 that allows, among other functions:

- Writing the MIMO process 2x2 to be analyzed.
- Calculating the RGA to measure any interaction between the process variables.
- Selecting the desired control structure.
- Designing the decoupling for centralized control structures.
- Tuning the various PID controllers in several MIMO control structures through genetic algorithm-based optimization.
- Establishing fixed values, if desired, for the controllers' parameters, or limiting their range of variation.
- Displaying the response of the selected control structure graphically.
- Saving the results obtained in a data file for later analysis.

- Comparing the responses of the various structures tested both graphically and numerically (fitness function).

This paper is organized as follows. Section II explains the control structures and tuning methods that can be applied with the tool. Section III gives a detailed description of the tool's functions. Section IV provides an evaluation of the tool, detailing the educational experience with the tool. Finally, Section V draws conclusions.

II. CONTROL STRUCTURES AND OPTIMIZATION ALGORITHMS USED

This section describes the different control structures for MIMO control systems 2x2 implemented in the developed tool and the optimization algorithm used.

A. MIMO Control Structures

All the control structures included in the developed tool are based on PID controllers, as these types of controllers are used in most industrial applications due their robustness, the intuitive relationship between their parameters and the response of the system, and their flexibility [14].

- *Decentralized Control* is performed with two PID controllers. A controller is assigned for each loop of a MIMO process 2x2, Fig. 1.

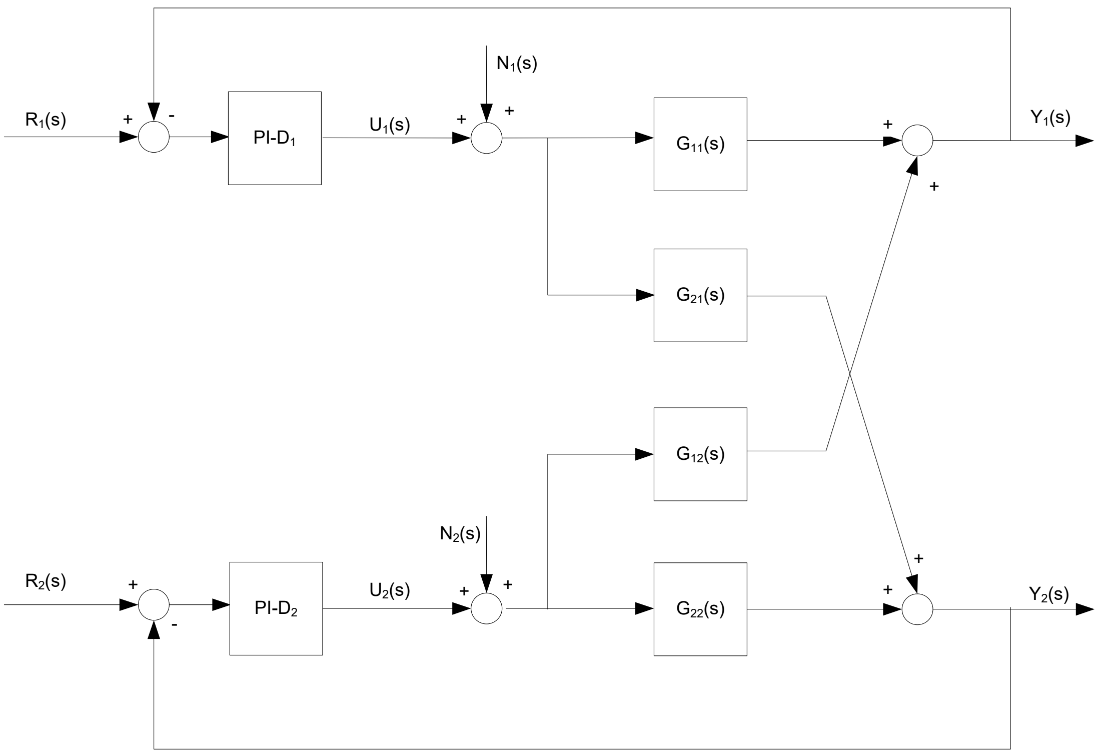


Fig. 1. Decentralized Control.

- Two alternatives to *Centralized Control* are considered: the use of a block made up of four PID controllers and a decentralized control made up of two PIDs, but with the addition of decoupling.
  - For a *centralized control case made up of four PIDs*, Fig. 2, two PIDs receive the error signal from one loop and the other two from the other loop ($e_1 = r_1 - y_1$ and $e_2 = r_2 - y_2$). Furthermore, this block provides two control signals to the process (u_1 and u_2), where $u_1 = u_{11} + u_{12}$ comes from PID_{11} PID_{12} and $u_2 = u_{21} + u_{22}$ comes from PID_{22} and PID_{21}.

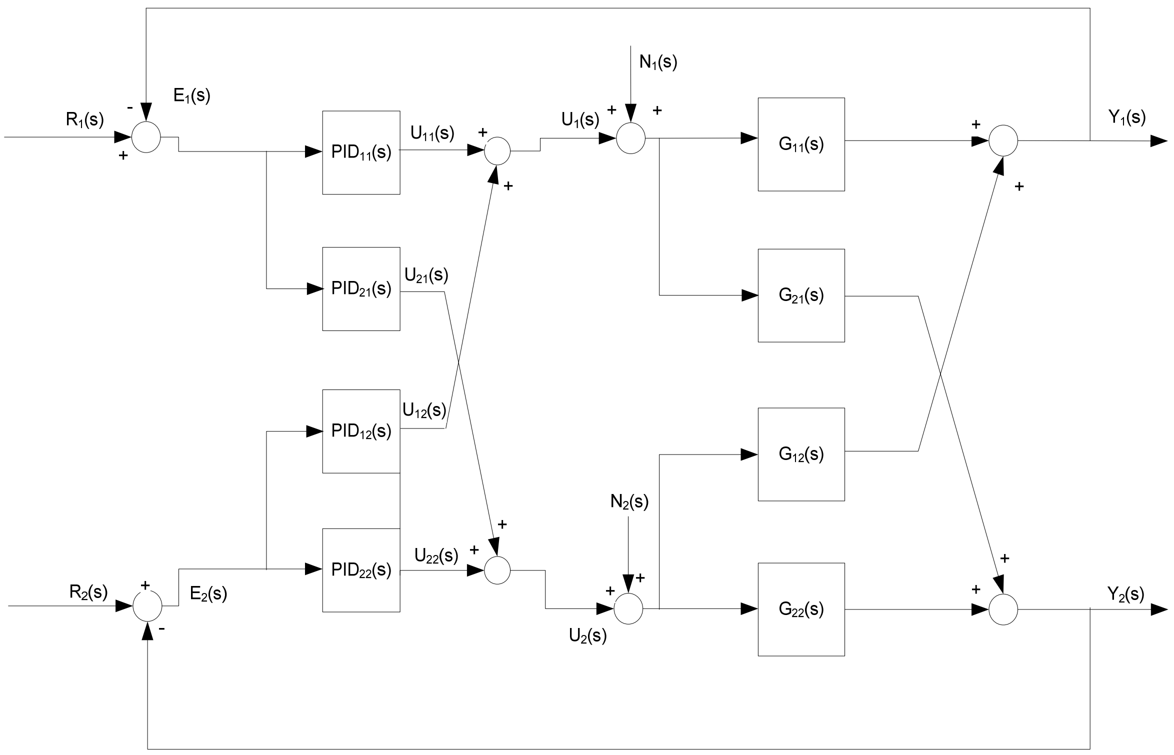


Fig. 2. Centralized Control.

- For the *centralized control with decoupling*, Fig. 3, a block is added to eliminate the process interaction, reducing the problem to one of decentralized control. Compared with other centralized structures, from a computational point of view, by reducing the number of controllers, the number of decision variables that the optimization algorithm has to handle later is also reduced, so this structure is more efficient than others. However, if the interaction is not completely eliminated, the results of the control design will not be those desired. There are different techniques to design the decoupling [13]. The tool offers a choice of the three most commonly used: Nordfeldt dynamic decoupling, Nordfeldt static decoupling and decoupling with two dynamic elements [14]. It also allows a user to write a custom decoupling manually.

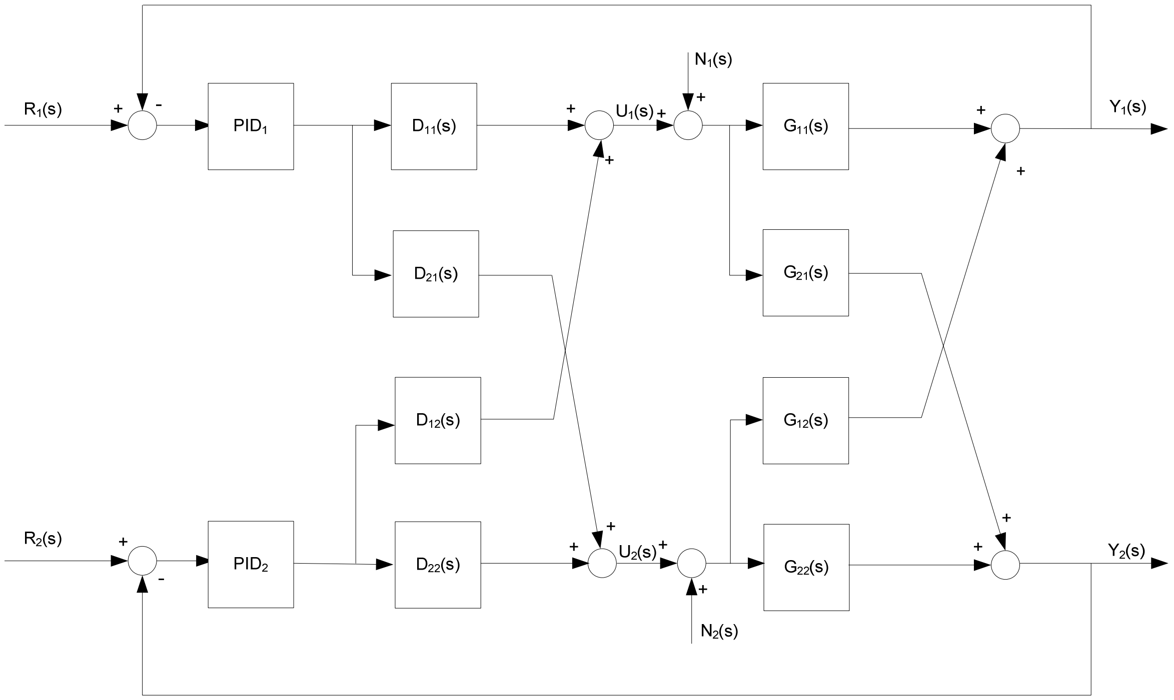


Fig. 3. Centralized Control by Decoupling.

The tool offers four options for PID controllers to be implemented in control structures [15]: PID, PI-D, I-PD or, simply, a PI controller.

B. Optimal Tuning by Genetic Algorithms

To tune the PID controller parameters, an optimization algorithm based on an evolutionary strategy was used, specifically a genetic algorithm (GA). The GA or bio-inspired algorithms are stochastic global optimization strategies that attempt to emulate natural evolution [16]. To find the optimal solution, operators such as crossover, mutation and selection [17] on solutions (individuals) obtained in previous stages of the algorithm are used. A stochastic algorithm cannot guarantee optimal solutions, but the experience of other authors in various fields has shown that the solutions provided by these strategies are very close to the optimum, with a low computational cost compared to deterministic algorithms [18].

The GA used is the Augmented Lagrangian Genetic Algorithm (ALGA) [19]. This GA can solve a Non-Linear optimization Problem (NLP) with equality and inequality constraints, and bounds in decision variables. The GA parameters, those which had given better performance, were configured to be:

- Population: The algorithm begins by creating a random initial population using a uniform distribution. Its size is 20 individuals in each generation, but the number of elements (genetic code) that characterize each of them depends on the MIMO control strategy chosen:
    - For decentralized control and centralized control by decoupling, there are eight elements per individual that represent the parameters of the two PIDs used (K_{p1}, T_{i1}, T_{d1}, α_1, K_{p2}, T_{i2}, T_{d2} y α_2).
    - For centralized control, there are sixteen elements per individual, that represent the four PID parameters involved (K_{p11}, T_{i11}, T_{d11}, α_{11}, K_{p12}, T_{i12}, T_{d12}, α_{12}, K_{p21}, T_{i21}, T_{d21}, α_{21}, K_{p22}, T_{i22}, T_{d22} y α_{22}).
- Individual coding: made by real numbers.
- Genetic operators:
    - Selection: the selection mechanism of parents for the next generation is a stochastic uniform distribution [19].

  - Crossover: The recombination mechanism or crossover begins with the creation of a random binary vector of the same length as the parent's genetic code. An algorithm runs through the vector so that, if it finds a 1, the corresponding gene is taken from the first parent, otherwise if it finds a 0; the gene is taken from the second parent. It was established that 80% of individuals of each generation come from this mechanism.
  - Mutation: an algorithm is employed that generates random address mutation, whose step length is chosen in order to verify the problem constraints. It has been established that 20% of individuals of each generation come from this mechanism.
  - Elite count: the existence of elite individuals is considered, so the best parents of each generation are moved directly to the next. Its value was set to two.
- Stopping Conditions: the next two are used, finishing the implementation of the GA at the time to verify either. The user can configure these values with the tool:
  - The point when new generations do not improve the fitness function.
  - The point at which the maximum number of generations configured in the tool has been reached.

Another important issue to consider is the construction of the fitness function to be optimized. The aim was to minimize both errors between the setpoint and process output as the control effort required. As the intention was to work with 2x2 processes, the following fitness function (J), made up of two terms (J_1 and J_2), was established:

$$J = J_1 + J_2 \tag{1}$$

$$J_1 = WeightIAE_1(r_1) * IAE_1(r_1) + WeightIAE_1(p) * IAE_1(p) + WeightTV_1 * TV_1 \tag{2}$$

$$J_2 = WeightIAE_2(r_2) * IAE_2(r_2) + WeightIAE_2(p) * IAE_2(p) + WeightTV_2 * TV_2 \tag{3}$$

Where:

- *$IAE_i(r_i)$*: the Integral of Absolute value of Error produced in output *i* by setpoint *r_i*.
- *$IAE_i(p)$*: Integral of Absolute value of Error produced in output *i* by external disturbances and the other setpoint.
- *TV_i*: the total variance, which quantifies the total control effort by measuring the evolution of the control signal u_i at each sampling instant *k* with respect to the instant before [20]:

$$TV_i = \sum_{k=1}^{\infty} \left| u_i(k) - u_i(k-1) \right| \tag{4}$$

- *$WeightIAE_i(r_i)$, $WeightIAE_i(p)$ y $WeightTV_i$*: weights that permit greater or lesser relevance to be assigned to certain terms of the fitness function, depending on the desired objectives (a more or less aggressive responses, for example).

III. TOOL DESCRIPTION

The following describes in detail the tool’s capabilities, designed for the optimal tuning of PID controllers in the control structures described above. To that end, a Twin Rotor MIMO System was used. The principal goal was not only to arrive at the designs themselves, but also to develop a tool sufficiently educational to be useful to students and to allow them to assimilate the basic knowledge of these kinds of processes.

To tune the controllers optimally for each control structure, the user has the option to configure either a single design, or a configuration of several designs that run sequentially using a batch process. Both options are described here.

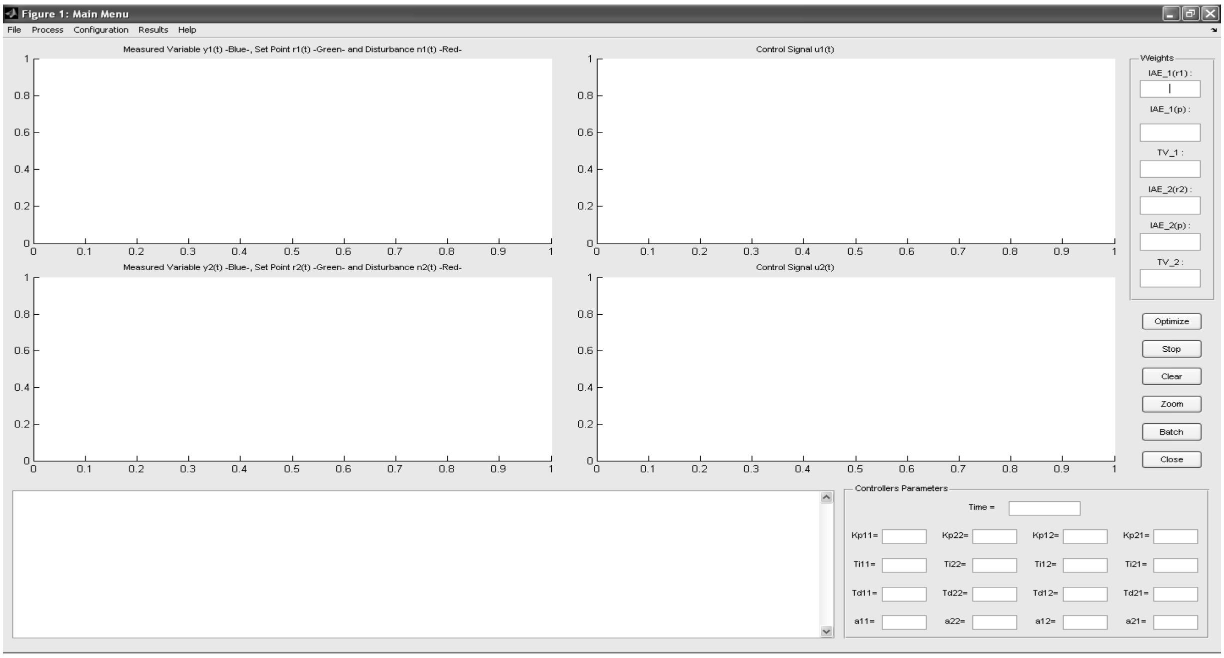


Fig. 4. Main Window.

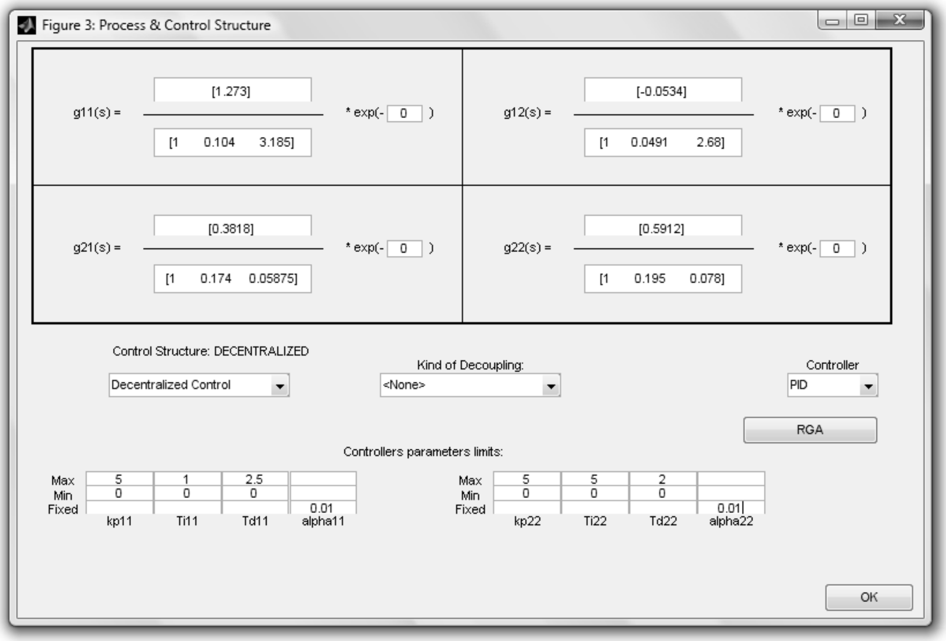
A. Configuration and Running of a Single Design

The model of the system used as an example was obtained by identification. Its mathematical expression is:

$$G(s)=\begin{pmatrix} \dfrac{1.273}{s^2+0.104s+3.185} & \dfrac{-0.05354}{s^2+0.0491s+2.68} \\ \\ \dfrac{0.3818}{s^2+0.174s+0.05875} & \dfrac{0.5912}{s^2+0.195s+0.078} \end{pmatrix} \qquad (5)$$

The steps to set up the design for a decentralized system control using the PID structure for the controllers are:

- **Enter the MIMO process 2x2**: click on the *Process menu* located on the Main Window, Fig. 4, which opens the *Process and Control Structure dialog box*, Fig. 5. At the top of this window are boxes to insert the four transfer functions for the process.



g11(s) = [1.273] / [1 0.104 3.185] * exp(- 0)

g12(s) = [-0.0534] / [1 0.0491 2.68] * exp(- 0)

g21(s) = [0.3818] / [1 0.174 0.05875] * exp(- 0)

g22(s) = [0.5912] / [1 0.195 0.078] * exp(- 0)

Control Structure: DECENTRALIZED

Decentralized Control

Kind of Decoupling: <None>

Controller: PID

Controllers parameters limits:

	kp11	Ti11	Td11	alpha11
Max	5	1	2.5	
Min	0	0	0	
Fixed				0.01

	kp22	Ti22	Td22	alpha22
Max	5	5	2	
Min	0	0	0	
Fixed				0.01

Fig. 5. Process and Control Structure Window.

- **Analyze the interaction between unpaired variables using the RGA**: In the previous dialog box the *RGA button* can be seen, Fig. 5. Press this button; a new window will appear showing the RGA of the process.
- **Select the control structure**: the desired control structure can be chosen in the *Process and Control Structure dialog box*, Fig. 5, using the *Control Structure pop-up menu*, and selecting either a centralized control by four PIDs or a decentralized control. If the latter is chosen, another pop-up menu (*Kind of Decoupling*) appears, where the user can choose whether or not to use decoupling, and if so, of what kind. In this example, a decentralized control without decoupling was chosen.

- **Select the PID control structure**: from the *Controller pop-up menu* in the dialog box, Fig. 5, one of the four PID control structures allowed by the tool (PID, PI-D, I-PD and PI) can be selected. In this case a PID structure was chosen.
- **Limit the variation range of controller parameters**: the last configurable field of *Process and Control Structure dialog box* is the search range used by the optimization algorithm for the decision variables (controller parameters). If desired, fixed values can be set. In this example, the filter on the derivative action factor is fixed to 0.01 and the range of other decision variables was selected after performing multiple tests.
- **Set simulation and optimization parameters**: the *Configuration menu* of the *Main Window*, Fig. 4, is used to select a set of parameters which can be divided into two groups:
    - *Simulation parameters*, Fig. 6a: Simulations must be performed both during and after each optimization. The post-optimization simulation is performed to show graphically how the system responds with the final control strategy. In this design the parameters shown in Table I were set.

TABLE I. Simulation Parameters

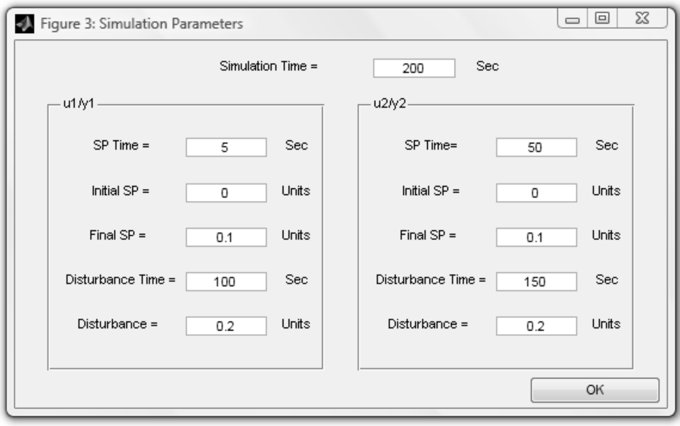
SIMULATION PARAMETERS	***U_1/Y_1***	***U_2/Y_2***
SP time: moment at which Set Point is produced	5 sec.	50 sec.
Initial SP: Set Point initial value	0 units	0 units
Final SP: Set Point final value	0.1 units	0.1 units
Disturbance Time: moment at which disturbance is produced.	100 sec.	150 sec.
Disturbance: step amplitude that simulates the disturbance.	0.2 units	0.2 units

- *Optimization parameters*, Fig. 6b: these are necessary to establish the values of some GA parameters, previously discussed in Section II.B. The duration times of the simulations to be performed during optimization are also set. In this case the next optimization parameters set were those shown in Table II.

TABLE II. Optimization Parameters

OPTIMIZATION PARAMETERS

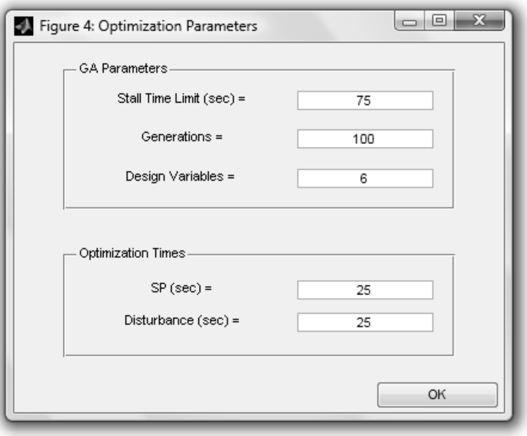
GA Parameters	*Stall time limit*	75 sec.
	Generations	100
	Design variables	6
Optimization Times	*SP*	25 sec.
	Disturbance	25 sec.



Simulation Time = 200 Sec

u1/y1
SP Time = 5 Sec
Initial SP = 0 Units
Final SP = 0.1 Units
Disturbance Time = 100 Sec
Disturbance = 0.2 Units

u2/y2
SP Time= 50 Sec
Initial SP = 0 Units
Final SP = 0.1 Units
Disturbance Time = 150 Sec
Disturbance = 0.2 Units



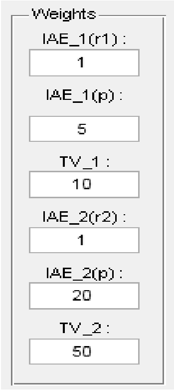
GA Parameters
Stall Time Limit (sec) = 75
Generations = 100
Design Variables = 6

Optimization Times
SP (sec) = 25
Disturbance (sec) = 25

Fig. 6a: Simulation Parameters (Left).

Fig. 6b: Optimization Parameters (Right).

- **Select the weights of the fitness function**: These weights are described in Section II.B, and are entered in the *Weights panel* of the Main Window, Fig. 7.



IAE_1(r1) : 1
IAE_1(p) : 5
TV_1 : 10
IAE_2(r2) : 1
IAE_2(p) : 20
TV_2 : 50

Fig. 7. Weights panel of the Main Window.

- **Write the name of the file in which the results will be saved**: in the dialog box that appears when the user clicks on the *File menu* in the *Main Window*.
- **Start optimization**: click the *Optimize button* in the *Main Window* to start the optimal tuning of the controllers.

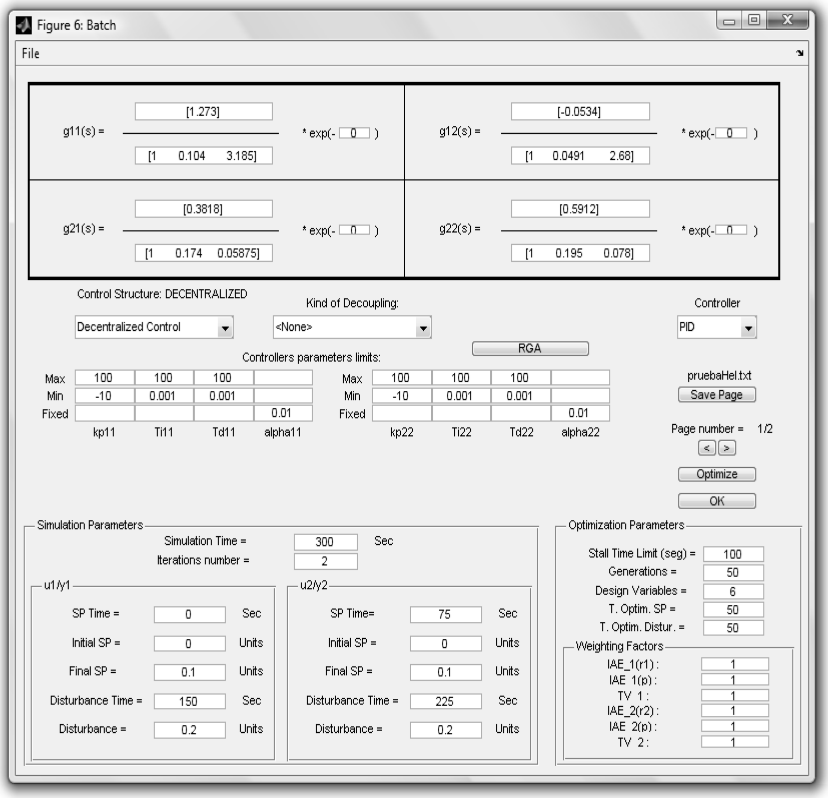
During the optimization, in the text box at the bottom of the Main Window information will appear on the progress of the optimization, that is, on the GA evolution in each generation and values of the fitness function.

When the optimization is finished, the controlled variable $y_1(t)$, the set point $r_1(t)$ and the disturbance at the input of the process $G_{11}(s)$ will be shown on the axes at the top left of the Main Window. On the axes at the bottom left the variable $y_2(t)$, the set point $r_2(t)$ and disturbance at input of process $G_{22}(s)$ will be seen. In the upper right hand axes the control signal $u_1(t)$, and in the axes at the bottom right the control signal $u_2(t)$, will appear. Also, in the *Controller Parameters panel* (bottom right) the values that were obtained for each of the PID parameters of the control strategy selected will be seen.

B. Configuration and Running of Batch Designs

To set up a series of designs that run sequentially the batch processing feature of the tool can be used; this is especially useful for unattended operation. Pressing the *Batch button* in the Main Window causes the *Batch dialog box*, Fig. 8, to appear. In this dialog box, using the *Iterations Number box* in the *Simulation Parameters* panel, the user can select the same options as those for the single design (introduction of the process, selection of the control structure and kind of PID, setting search ranges of controller parameters, setting the parameter optimization and simulation and the weights of fitness function) and selecting the number of times that he/she wishes to make the same design (an important issue, taking into account the stochastic nature of the optimization algorithm). In the example described in this paper a batch design, Fig. 8, was programmed. The process, the structure control and parameters used are the same as those explained in Section III.A.1. To program different designs for different processes with different control structures and controllers, with other optimization and simulation parameters, and so on, different pages can be generated. In this case, different pages were created in which the only differences were the weights of the fitness function.

Finally, the *File menu* in the *Batch dialog box,* Fig. 8, can be used to retrieve a previously programmed batch file. If selected, the name of the file and the first page of data will appear in the Batch dialog box.



g11(s) = [1.273] / [1 0.104 3.185] * exp(- 0)
g12(s) = [-0.0534] / [1 0.0491 2.68] * exp(- 0)
g21(s) = [0.3818] / [1 0.174 0.05875] * exp(- 0)
g22(s) = [0.5912] / [1 0.195 0.078] * exp(- 0)

Control Structure: DECENTRALIZED
Decentralized Control
Kind of Decoupling: <None>
Controller: PID

Controllers parameters limits:

	kp11	Ti11	Td11	alpha11	kp22	Ti22	Td22	alpha22
Max	100	100	100		100	100	100	
Min	-10	0.001	0.001		-10	0.001	0.001	
Fixed				0.01				0.01

pruebaHel.txt
Page number = 1/2

Simulation Parameters
Simulation Time = 300 Sec
Iterations number = 2

	u1/y1		u2/y2	
SP Time =	0	Sec	75	Sec
Initial SP =	0	Units	0	Units
Final SP =	0.1	Units	0.1	Units
Disturbance Time =	150	Sec	225	Sec
Disturbance =	0.2	Units	0.2	Units

Optimization Parameters
Stall Time Limit (seg) = 100
Generations = 50
Design Variables = 6
T. Optim. SP = 50
T. Optim. Distur. = 50

Weighting Factors
IAE_1(r1) : 1
IAE 1(p) : 1
TV 1 : 1
IAE_2(r2) : 1
IAE 2(p) : 1
TV 2 : 1

Fig. 8. Batch dialog box.

C. Analysis of Results

As discussed above, the results are saved in a file. In this example a batch processing has been programmed, so that in the saved file there will be a line of results for each programmed page.

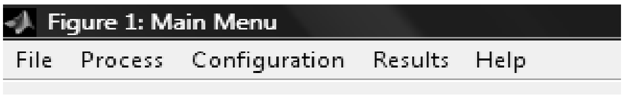
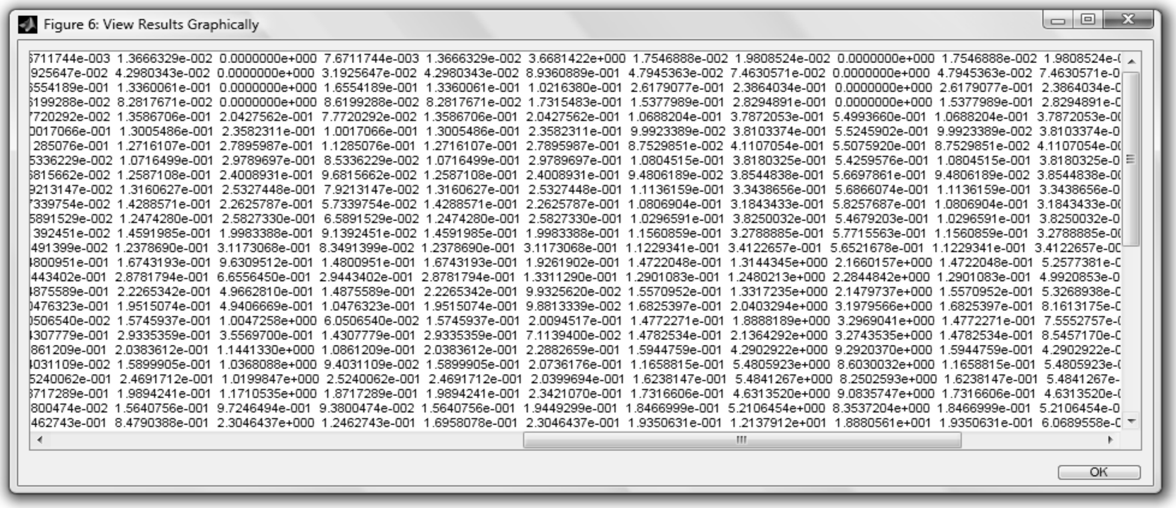


Fig. 9. Menus available from the Main Menu

Pressing the *Results menu* on the *Main Window*, Fig. 9, the dialog box shown in Fig. 10 will open, showing all the results stored in the file. Also, the *Selection Process and Control Structure dialog box*, explained above, will appear. Here the control system, the control structure and PID controller can be seen. Furthermore, the values that were selected for the simulation and optimization parameters will appear in the appropriate windows destined, Fig. 6), as previously explained.

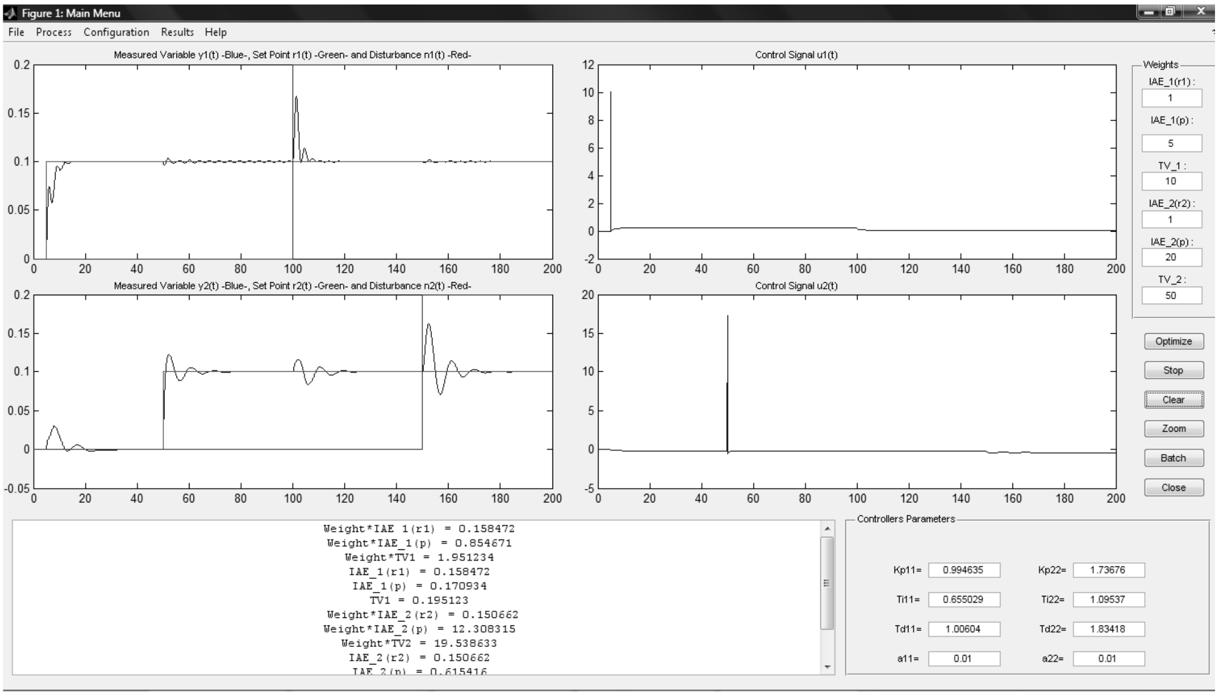


```
6711744e-003 1.3666329e-002 0.0000000e+000 7.6711744e-003 1.3666329e-002 3.6681422e+000 1.7546888e-002 1.9808524e-002 0.0000000e+000 1.7546888e-002 1.9808524e-
925647e-002 4.2980343e-002 0.0000000e+000 3.1925647e-002 4.2980343e-002 8.9360889e-001 4.7945363e-002 7.4630571e-002 0.0000000e+000 4.7945363e-002 7.4630571e-0
6554189e-001 1.3360061e-001 0.0000000e+000 1.6554189e-001 1.3360061e-001 1.0216380e-001 2.6179077e-001 2.3864034e-001 0.0000000e+000 2.6179077e-001 2.3864034e-
6199288e-002 8.2817671e-002 0.0000000e+000 8.6199288e-002 8.2817671e-002 1.7315483e-001 1.5377989e-001 2.8294891e-001 0.0000000e+000 1.5377989e-001 2.8294891e-
7720292e-002 1.3586706e-001 2.0427562e-001 7.7720292e-002 1.3586706e-001 2.0427562e-001 1.0688204e-001 3.7872053e-001 5.4993660e-001 1.0688204e-001 3.7872053e-0
0017066e-001 1.3005486e-001 2.3582311e-001 1.0017066e-001 1.3005486e-001 2.3582311e-001 9.9923389e-002 3.8103374e-001 5.5245902e-001 9.9923389e-002 3.8103374e-0
285076e-001 1.2716107e-001 2.7895987e-001 1.1285076e-001 1.2716107e-001 2.7895987e-001 8.7529851e-002 4.1107054e-001 5.5075920e-001 8.7529851e-002 4.1107054e-0
5336229e-002 1.0716499e-001 2.9789697e-001 8.5336229e-002 1.0716499e-001 2.9789697e-001 1.0804515e-001 3.8180325e-001 5.4259576e-001 1.0804515e-001 3.8180325e-0
6815662e-002 1.2587108e-001 2.4008931e-001 9.6815662e-002 1.2587108e-001 2.4008931e-001 9.4806189e-002 3.8544838e-001 5.6697861e-001 9.4806189e-002 3.8544838e-0
9213147e-002 1.3160627e-001 2.5327448e-001 7.9213147e-002 1.3160627e-001 2.5327448e-001 1.1136159e-001 3.3438656e-001 5.6866074e-001 1.1136159e-001 3.3438656e-0
7339754e-002 1.4288571e-001 2.2625787e-001 5.7339754e-002 1.4288571e-001 2.2625787e-001 1.0806904e-001 3.1843433e-001 5.8257687e-001 1.0806904e-001 3.1843433e-0
5891529e-002 1.2474280e-001 2.5827330e-001 6.5891529e-002 1.2474280e-001 2.5827330e-001 1.0296591e-001 3.8250032e-001 5.4679203e-001 1.0296591e-001 3.8250032e-0
392451e-002 1.4591985e-001 1.9983388e-001 9.1392451e-002 1.4591985e-001 1.9983388e-001 1.1560859e-001 3.2788885e-001 5.7715563e-001 1.1560859e-001 3.2788885e-0
491399e-002 1.2378690e-001 3.1173068e-001 8.3491399e-002 1.2378690e-001 3.1173068e-001 1.1229341e-001 3.4122657e-001 5.6521678e-001 1.1229341e-001 3.4122657e-0
4800951e-001 1.6743193e-001 9.6309512e-001 1.4800951e-001 1.6743193e-001 1.9261902e-001 1.4722048e-001 1.3144345e+000 2.1660157e+000 1.4722048e-001 5.2577381e-
443402e-001 2.8781794e-001 6.6556450e-001 2.9443402e-001 2.8781794e-001 1.3311290e-001 1.2901083e-001 1.2480213e+000 2.2844842e+000 1.2901083e-001 4.9920853e-0
4875589e-001 2.2265342e-001 4.9662810e-001 1.4875589e-001 2.2265342e-001 9.9325620e-002 1.5570952e-001 1.3317235e+000 2.1479737e+000 1.5570952e-001 5.3268938e-
0476323e-001 1.9515074e-001 4.9406669e-001 1.0476323e-001 1.9515074e-001 9.8813339e-002 1.6825397e-001 2.0403294e+000 3.1979566e+000 1.6825397e-001 8.1613175e-
0506540e-002 1.5745937e-001 1.0047258e+000 6.0506540e-002 1.5745937e-001 2.0094517e-001 1.4772271e-001 1.8888189e+000 3.2969041e+000 1.4772271e-001 7.5552757e-
4307779e-001 2.9335359e-001 3.5569700e-001 1.4307779e-001 2.9335359e-001 7.1139400e-002 1.4782534e-001 2.1364292e+000 3.2743535e+000 1.4782534e-001 8.5457170e-
861209e-001 2.0383612e-001 1.1441330e+000 1.0861209e-001 2.0383612e-001 2.2882659e-001 1.5944759e-001 4.2902922e+000 9.2920370e+000 1.5944759e-001 4.2902922e-
4031109e-002 1.5899905e-001 1.0368088e+000 9.4031109e-002 1.5899905e-001 2.0736176e-001 1.1658815e-001 5.4805923e+000 8.6030032e+000 1.1658815e-001 5.4805923e-
5240062e-001 2.4691712e-001 1.0199847e+000 2.5240062e-001 2.4691712e-001 2.0399694e-001 1.6238147e-001 5.4841267e+000 8.2502593e+000 1.6238147e-001 5.4841267e-
8717289e-001 1.9894241e-001 1.1710535e+000 1.8717289e-001 1.9894241e-001 2.3421070e-001 1.7316606e-001 4.6313520e+000 9.0835747e+000 1.7316606e-001 4.6313520e-
800474e-002 1.5640756e-001 9.7246494e-001 9.3800474e-002 1.5640756e-001 1.9449299e-001 1.8466999e-001 5.2106454e+000 8.3537204e+000 1.8466999e-001 5.2106454e-0
462743e-001 8.4790388e-001 2.3046437e+000 1.2462743e-001 1.6958078e-001 2.3046437e-001 1.9350631e-001 1.2137912e+001 1.8880561e+001 1.9350631e-001 6.0689558e-
```

Fig. 10. Results Window.

If a line of results (containing the process analyzed, the control structure tested, the PID parameters and the values of fitness function) is selected and the *OK button* is pressed, a representation of the process response with the control structure obtained will appear in axes on the *Main Window*. Fig. 11 shows the response obtained for the example studied. Looking at the axes it can be seen that outputs $y_1(t)$ and $y_2(t)$ reach their reference values and reject disturbances in quite an optimal way. Also, in the *text box* the values of the different elements of the fitness function achieved with this test will appear. The values obtained in the example shown are the lowest of all those achieved, which was why this particular result line was selected to be analyzed. In the *Weights Panel* the weights selected for this case will appear and in the *Controller Parameters*

Panel the values of the PID parameters obtained can be seen. If more than one result line is selected, a graph of the results will be shown for each of the lines, allowing an intuitive and visual comparison of the different control structures obtained.



Weights

IAE_1(r1) : 1

IAE_1(p) : 5

TV_1 : 10

IAE_2(r2) : 1

IAE_2(p) : 20

TV_2 : 50

```
Weight*IAE 1(r1) = 0.158472
Weight*IAE_1(p) = 0.854671
Weight*TV1 = 1.951234
IAE_1(r1) = 0.158472
IAE_1(p) = 0.170934
TV1 = 0.195123
Weight*IAE_2(r2) = 0.150662
Weight*IAE_2(p) = 12.308315
Weight*TV2 = 19.538633
IAE_2(r2) = 0.150662
IAE_2(p) = 0.615416
```

Controllers Parameters

Kp11= 0.994635

Ti11= 0.655029

Td11= 1.00604

a11= 0.01

Kp22= 1.73676

Ti22= 1.09537

Td22= 1.83418

a22= 0.01

Fig. 11. Main Window

D. Other Options

On the Main Window, Fig. 10, there are a number of buttons under the *Weights Panel*, listed below with their corresponding functions:

- *Optimize*: starts the optimization.
- *Stop*: stops the optimization at any time.
- *Clear*: clears the contents of graphic axes.
- *Zoom*: enlarges the area required in the axes.

- *Batch*: pressing this button opens a dialog box that permits a batch optimization to be generated.
- *Close*: closes the tool.

IV. EVALUATION

The main goal in creating this tool was to improve student learning in designing control systems for multivariable processes. The tool was therefore put into service for the practical work in the course Control Engineering, in the Automatic Control Engineering degree at the Polytechnic School of the University of Cordoba, Spain. When all the students had finished their practicals and had been graded by the teacher they were asked to express their anonymous opinion of the tool, with the aim of analyzing the effect that it had had on their learning.

A. Student Practicals With the Tool

In the practical work of the Control Engineering course, students had to make various designs and then compare the results obtained by simulation with the results obtained on a testing platform in the laboratory, Feedback's Twin Rotor MIMO System 33-220, Fig. 12, a classic example of a MIMO process 2x2. The process comprises two perpendicular rotors for which independent voltages (inputs) are supplied with the objective of controlling the angular positions of the two axes on the horizontal and vertical planes (outputs).

One of the best possible solutions from all those obtained by the students for the control system platform is the example analyzed in previous sections.

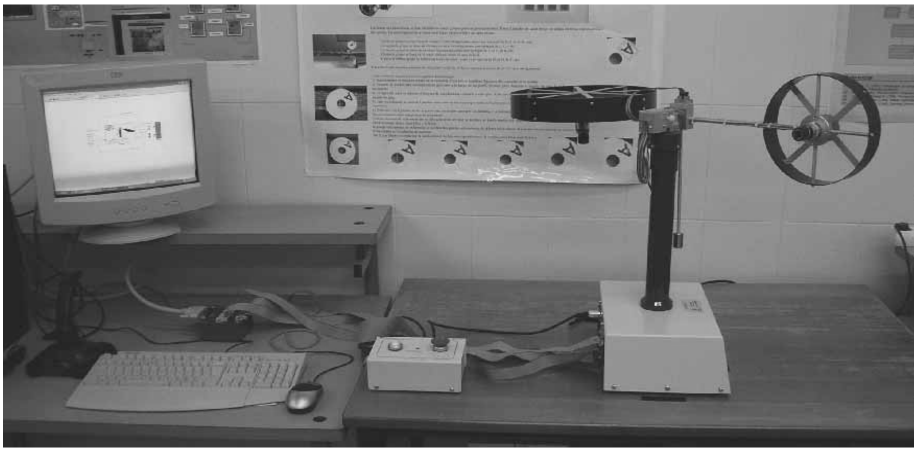


Fig. 12. Twin Rotor MIMO System 33-220 of Feedback.

After completing their practical work, students were required to deliver a reasoned and justified report with the results obtained. The subsequent evaluation of these reports showed that they had attained the high level of understanding of concepts that it was intended to teach. The improvement in the practical grades over the previous year can be seen in Fig. 13, and is attributed to the tool allowing students to check in practice the knowledge they had received in their lectures. This tool allows students to design all the control structures for MIMO processes that had been taught in class, to test and compare their operation by changing certain parameters, and to see what option is best for the process being studied through the tool's analysis capabilities. The tool's GUI allows simulations to be run without any programming effort on the part of the students, who thus improve their understanding without spending time on implementing the controllers.

Fig. 13. Comparison of student grades between 2008/2009 and 2009/2010 courses

A statistical analysis was made on the student results [21]. Statistical outcomes of this study are shown in Table III. The mean of the grades in the 2009/2010 offering of the academic course in which the tool was used in practicals is higher than of the previous one. To determine if this difference between means is statistically significant a t-test was made, giving a t-test value of 2.2267; choosing in a usual t-test table a value of 0.05 for probability p the critical t value is 2.04 (less than 2.2267), thus the means are statistically different. Students having improved their grades significantly support the utility of the tool to improve student learning.

Table III. Descriptive Statistical Values

	2008/2009 course	2009/2010 course
Mean	5.64	7.84
Standard Error	0.5174	0.4632
Median	6	8.375
Minimum	2.25	3.5
Maximum	9.25	9.75
Count	18	16
Standard Deviation	2.195	1.8526
Sample Variance	4.818	3.432

B. Student Assessment of the Educational Tool

To evaluate three other basic aspects of the tool in student learning, the questionnaire shown in Table IV was administered. The questions were formulated based on those used in similar studies [12], [13], [21]-[23].

TABLE IV. Survey of tool carried out for the students

	Item
Q1	The tool helped me to improve my theoretical knowledge about MIMO control systems.
Q2	I think that I will remember the concepts taught in the subject better than if I had only had lecture classes.
Q3	I understood the design methodology that is used with the tool correctly and easily.
Q4	I am satisfied with the simulation practical with the tool.
Q5	The concepts presented in the tool are clear and easy to follow.
Q6	I think that the tool is easy to use and understand.
Q7	I think that the Graphical Interface User is user-friendly.

The first three items were proposed to elicit students' opinions about the improvement in their learning when using the tool. The next item was included to determine if they considered that the practical achieved its objective. The last three questions relate to the usability and ease of understanding of the tool. In Table V the responses of the students are given.

TABLE V. Student Responses to the tool survey per subscale (Number of students= 16)

Group Items	**Strongly Agree**	**Agree**	**Neutral**	**Disagree**	**Strongly Disagree**
Improvement in their learning (Q1, Q2, Q3)	41.7%	50%	8.3%	0	0
Objective practice (Q4)	25%	75%	0	0	0
Usability and easy understanding of the tool (Q5, Q6, Q7)	37.5%	54.2%	8.3%	0	0

Fig. 14 details the survey responses of all 16 students enrolled during academic year 2009/2010, who did the practical. The answers were rated as *strongly agree*, *agree*, *neutral*, *disagree*, or *strongly disagree.* The percentage of answers that strongly agree or agree with the assertions is very high, indicating their opinions that using the tool in the practicals served to consolidate the concepts taught in the course, and that the tool is user friendly for most of them. Therefore, it would seem to be of interest to continue this experiment and to prepare more practicals with this tool in future.

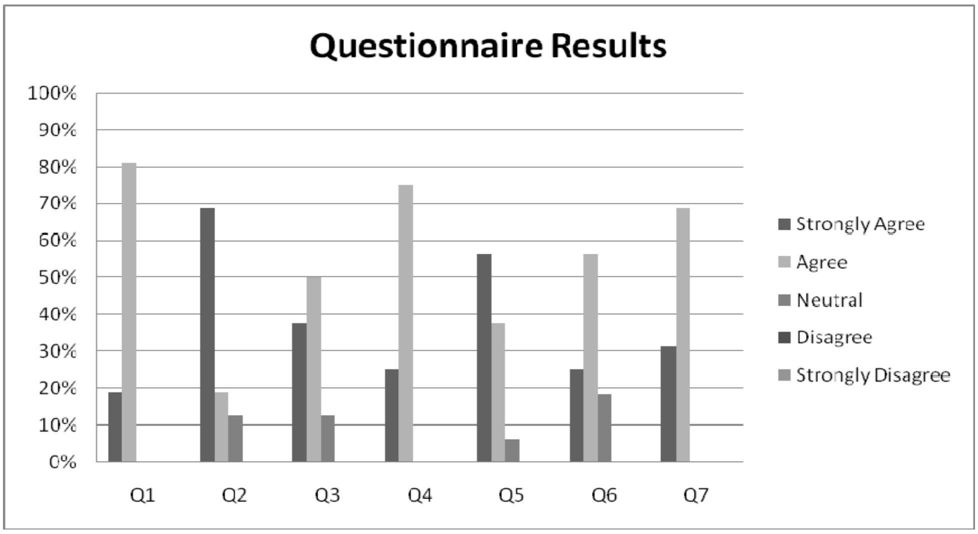


Fig. 14. Student survey answers (Number of students= 16)

V. CONCLUSIONS AND FUTURE IMPROVEMENTS

This paper has presented a software tool for optimal tuning of 2x2 MIMO control systems based on Gas, that has a clear educational application. The tool has a friendly Graphical User Interface with many configuration options, for both design and analysis of results, including: calculation of the RGA; selection of different control structures; the decoupling design; the possible configuration of a single design or a batch design for tuning various types of PID controllers in various control structures by optimization based on genetic algorithms; graphical visualization of the response obtained with the selected control structure; storage of the results in data files for further analysis; and the comparison of responses obtained with the different structures tested both graphically, and numerically with the optimization data (fitness function). These features make it ideal for use in pedagogy, especially for control engineering practical laboratories. To test the usefulness of the developed tool, it was made available to students in practicals for them to obtain various 2x2 MIMO controller designs that were then compared and subsequently tested on a laboratory plant (Feedback's TRMS). The student response was satisfactory, both for the high degree of interest and for improving their practical grades.

Encouraged by these good results, it is proposed to incorporate future improvements to the tool, such as extending it to work with 3x3 MIMO systems, or to create a library of genetic algorithms which would allow the user to choose the most appropriate optimization algorithm.

ACKNOWLEDGEMENTS

This work was supported by the Spanish Ministerio de Ciencia e Innovación under grant DPI2007-62052. This support is very gratefully acknowledged.

REFERENCES

[1] W.L. Luyben, "*Practical Distillation Control",* Ed. W.L. Luyben. 1992

[2] F.G. Shinskey, "*Process Control System"*, Ed. MacGraw-Hill. 1995

[3] A. Niederlinski, “*A heuristic approach to the design of linear multivariable interacting control systems”*, Automatica 7, pp. 691-70, 1971

[4] M. Zhuang, D. Athertoon “*PID controllers design for a TITO system”*, IEE Proc. Control Theory Appl, vol. 141, no. 2, pp. 111-120, 1994.

[5] Y. Halevy, Z. Palmor, T. Efrati, ”*Automatic tuning of decentralized PID controllers for MIMO processes”*, J. Proc. Cont. Elsevier Science, Ltd. vol. 7, no. 2, pp 119-128, 1998.

[6] K. Toh, R. Davanathan, “*An expert autotuner for multiloop SISO controllers”,* Control Eng. Practice, vol. 1, no. 6 pp 999-1008, 1993.

[7] S.J. Shiu, S. Hwang, “*Sequential design method for multivariable decoupling and multiloop PID controllers”,* Ind. Eng. Chem. Res., vol. 37, no. 1, pp. 107-119, 1998.

[8] F. Vázquez, F. Morilla. “*Tuning Decentralized PID Controllers for MIMO Systems with Decouplers”*, in Proc. 15 IFAC World Congress, Pergamon (Elsevier Science), Barcelona, 2002.

[9] F. Vázquez, F. Morilla, S. Dormido, “*Entorno para Simulación, Análisis y Sintonía de Sistemas con Control Descentralizado 2x2”.* XIX Jornadas de Automatica, Madrid, 1998.

[10] M. Andersson, S. Gunnarsson, T. Glad, M. Norrlöf, “*A Simulation and AnimationTool For Studying Multivariable Control”,* in Proc. 15th IFAC Triennal World Congress, 2002.

[11] J, Albino Méndez, C. Lorenzo, L. Acosta, S. Torres, E. González, “*A Web-Based Tool For Control Engineering Teaching”,* Computer Applications in Engineering Education, vol. 14, no. 3, pp. 178-187, 2005.

[12] R. Dormido, H. Vargas, N. Duro, J. Sánchez, S. Dormido-Canto, G. Farias, F. Esquembre, S. Dormido, "*Development of a Web-Based Control Laboratory for Automation Technicians: The Three-Tank System",* IEEE Transactions on Education, vol. 51, no. 1, pp. 35-43, February 2008.

[13] J. Sánchez, S. Dormido, R. Pastor, F. Morilla, "*A Java/Matlab-Based Environment for Remote Control System Laboratories: Illustrated With an Inverted Pendulum",* IEEE Transactions on Education, vol. 47, no. 3, pp. 321-329, August 2004.

[14] F. Morilla García, "*Apuntes del curso de doctorado: Control Multivariable"*, Dpto. de Informática y Automática, UNED. 2008

[15] F. Morilla García, "*Apuntes PID*. Dpto. de Informática y Automática", Dpto. de Informática y Automática, UNED. 2008

[16] G. Vlachos, J.T. Evans, D. Williams, "*PI controller tuning for multivariable processes using genetic algorithms. Genetic Algorithms in Engineering Systems",* Innovations and Applications, Conference Publication No. 446, © IEE, September 1997, pp. 43-49.

[17] Ch. Runwei, "*Genetic algorithms and engineering optimization"*. 1ª ed. New York [USA]: Wiley-IEEE, p. 495, 2000.

[18] J.E. Jiménez-Hornero, "*Contribuciones al modelado y optimización del proceso de fermentación acética"*. Tesis Doctoral. Córdoba, 2007.

[19] A.R. Conn, N.I.M. Gould, P.L. Toint, "*A globally convergent augmented lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds"*. Mathematics of Computation, vol. 66, no. 217, pp 261-288, 1997.

[20] S. Skogestad, "*Simple analytic rules for model reduction and PID controller tuning"*, Journal of Process Control, vol. 13, pp 291-309, 2003.

[21] S. Maravic Cisar, R. Pinter, D. Radosav, P. Cisar, "*Software Visualization: the Educational Tool to Enhance Student Learning*", in Proc. 33th International Convection MIPRO, May 2010, pp. 990-994.

[22] Emmanuel A. Gonzalez, Martin Christian G. Leonor, Pauline Anne T. M. Mangulabnan, John Jessie S. Lu Chui Kau, Marlon Wilson U. Reyes, "*Work in Progress – An Educational Tool fo Teaching Linear and Control Systems*", 37th ASEE/IEEE Frontiers in Education Conference, Session T3J, October 2007, pp. 14-15.

[23] Marjan Mernik, Viljem Zumer, "*An Educational Tool for Teaching Compiler Construction*", IEEE Transaction on Education, Vol. 46, No. 1, pp.61-68, February 2003

BIOGRAPHIES

Daniel Carmona Morales received the M.S. degree in electronics and automatic control from Universidad de Cordoba (Spain) in 2009.

He is currently working with Relequick S. A. in Toledo, Spain, in the research and development department. He is interested in process modeling, simulation and control.

Jorge E. Jimenez-Hornero received the M.S. degree in electronics and automatic control engineering from Universidad de Córdoba (Spain) in 2000 and the Ph.D. degree in automatic control from Universidad Nacional de Educación a Distancia (UNED), Madrid (Spain), in 2007.

He is currently an associate professor in the Department of Computer Sciences and Automatic Control at Universidad de Córdoba. His major fields are control engineering, process optimization, modeling/simulation and teaching research on control engineering.

Dr. Jimenez-Hornero is member of the Spanish Committee of Automatic Control of the International Federation of Automatic Control (IFAC).

Francisco Vazquez was born in Cordoba (Spain) in 1968. He received the Dipl. Ing. degree in electronics in 1991, the Lic.es Sci in 1995 in Physical Sciences and the Ph.D. degree in automatic control from Universidad Nacional de Educación a Distancia (UNED), Madrid (Spain), in 2001.

He is currently working with Department of Computer Sciences and Automatic Control at Universidad de Cordoba (Spain) as an associate professor. Previously, he worked with UNED as a fellow student for four years. He is interested in control systems, multivariable control, decoupling techniques and advances methodologies of simulation and modeling, including objects oriented modeling.

Dr. Vazquez is nowadays the director of the School of Polytechnics Sciences at Universidad de Cordoba and belongs to Spanish Committee of Automatic Control of the International Federation of Automatic Control (IFAC).

Fernando Morilla received the degree in physics from Seville University, Seville, Spain, in 1979 and the Ph.D. degree from Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain, in 1987.

In 1983, he joined the Department of Computer Sciences and Automatic Control at the UNED where, since 1998, he is full professor of Control Engineering. He is interested in process modeling, simulation and control, PID control, multivariable control, control education and System Dynamics applications.

Dr. Morilla is member of the Spanish Committee of Automatic Control of the International Federation of Automatic Control (IFAC).