

Based on the Multi Variable System Design and Parameter Tuning Method of PID Controller

Li-Fei Deng¹, Yaowu Shi¹, Lan-Xiang Zhu^{2*}, D. L. Yu³ and Rui Zhu⁴

¹College of Communication Engineering, Jilin Univ., Changchun 132013, China

²Changchun Architecture And Civil engineering College, Changchun 130069, China

³Control Group, School of Engineering, Liverpool John Moores University, Liverpool, U.K

⁴Changchun zonbo new energy transmission system of limited liability company, China

Corresponding author: 519007262@qq.com

Abstract

This paper compares two approaches to tune PI controllers in a multivariable process. A modified Direct Synthesis method extends the technique to a multivariable system. Results show that the modified Direct Synthesis technique relies on model accuracy and suitability of the desired transfer function and process model. The effectiveness of this technique may break down when these conditions are not met. Genetic algorithms provide a method based on the evolution of species via natural selection. The GA method allows the use of a specific structure, such as PI controller, and then gets the best possible results for that structure. The concept has been adapted into an optimization technique that can be used to find the best controller parameters to meet user defined transient specifications for a complex process. Thus, the robustness of the method allows it to outperform the former technique. Simulation results are shown to illustrate the comparisons between these two methods and the effectiveness of Gas.

Keywords: Decentralized control, multivariable systems, PI controller tuning, genetic algorithm

1. Introduction

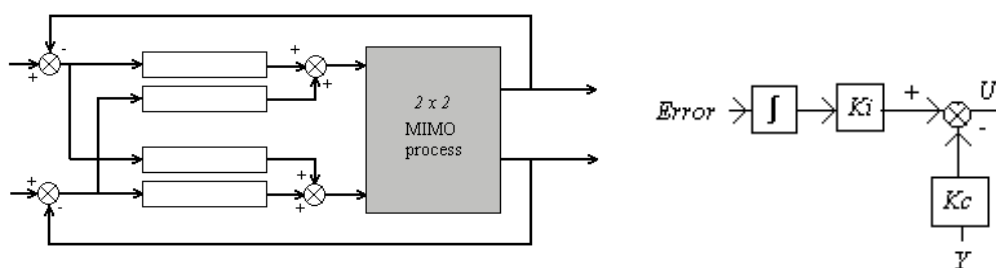
Although research on modern control theory has been thriving for many years and made significant theoretical progress, it is difficult to apply many of these theories in real practical problems. For example, these approaches may produce non-standard forms of controllers that make them difficult to implement and understand. Even when a suggested multivariable solution is theoretically good, it may not be easily implemented in engineering practice. Certain existing controller structures have been standardized and are still being used widely today. The Proportional and Integral (PI) controller is included in this category. Even though the task of tuning a PI controller is quite tedious and time consuming, the controller itself is easy to obtain and apply. There are a number of empirical tuning techniques available such as the Zeigler-Nichols (Z-N) [11] methods. These empirical methods do not require a process model and the tuning of the controller parameters are obtained experimentally. This method is time-consuming especially when there are several loops to consider. Also, the final controller parameters may not be realisable. Therefore, this method is not practical for many multivariable systems. However, Franke, Kruger and Knoop [3] developed an approach for a multivariable process using the DSM by substituting all the s-terms that defines the controller parameters. This allows the benefits of DSM to be applied to MIMO systems for a

specified controller structure. The feedback controller is designed by specifying the desired transient closed-loop response. This revised method also allows the controller structure to be fixed in a PI form. Figure1 illustrates a multivariable process with decentralized controllers for each loop in the system, known as the multiloop control structure [7]. However, the s-term and the desired transient transfer function values are limited within a certain acceptable range. This technique also depends on the accuracy of the system model.

Genetic Algorithms (GAs) provide an alternative method. They are global search methods that are based on natural population genetics and have been used as an optimizing tool in control systems. Holland [4] and De Jong [2] and others have demonstrated the excellent achievement of GAs. The powerful capabilities of genetic algorithms can be utilized to locate near optimum values of controller tuning parameters to meet an operator defined performance specification [8-9]. The required process performance will be specified in terms of output closed-loop transient responses. Controller parameters will be evaluated, by simulation, to meet the corresponding closed-loop system performance using an objective function, which can be user defined. Due to the robustness of this technique, the performance of GAs does not rely on the characteristic of the plant under control [10]. Thus, GAs is applicable to a wide range of practical plant. A decentralized PI controller structure is used for this paper. This controller structure is a desirable controller structure for a MIMO process because of its design simplicity [7], which is a good advantage in transferring the theoretical procedures to real life controller implementations.

2. Modified Direct Synthesis Method

In the general Direct Synthesis Technique, controller expressions can be obtained from the desired system output specification and if the process model is available. This design however, may not result in a standard P/PI/PID form and it is difficult to apply to a MIMO system and the final controller parameters obtained may not be realisable at all. Franke, Kruger and Knoop [3] uses a different approach of this DSM but still uses the system closed loop transfer function and defines the form of the controller structure required. The controller parameters are then obtained by substituting the s-term; in the controller parameters transfer functions, with a numerical value within a certain range.



(a) 2-Input 2-Output MIMO Process with PI Controllers (b) PI Controller Configuration for each Loop

Figure 1. Multivariable PI Control

Where:

K_c =Proportional controller parameter

K_I = Integral Action controller parameter

$U(s)$ =Controller output

The desired closed-loop transfer function can be obtained using the SISO or MIMO closed loop system as in Figure1 (a & b) [3].

$$Y(s) = G(s)U(s) \tag{1}$$

$$U(s) = \frac{1}{s} K_i R(s) - \frac{1}{s} K_i Y(s) - K_c Y(s) \tag{2}$$

Substitute U(s) into equation (1):

$$Y(s) = G(s) \left(\frac{1}{s} K_i R(s) - \frac{1}{s} K_i Y(s) - K_c Y(s) \right) \tag{3}$$

$$\frac{Y(s)}{R(s)} = F(s) \tag{4}$$

Where:

F(s) = Specified system closed-loop transfer function,

G(s) = Multivariable process transfer function,

Note that these equations involve matrices. Thus:

$$F(s) = \frac{Y(s)}{R(s)} = \left(I_p + G(s) \left(\frac{K_i}{s} + K_c \right) \right)^{-1} G(s) \frac{1}{s} K_i \tag{5}$$

Where:

p = number of inputs/outputs ;

I = identity matrix

Note that in equation (5), the desired transfer function F(s) is standardised, according to the user specifications. Rearranging this equation will lead to the controller in s- domain expression as in equation (6).

$$[K_c \ K_i] = -[G^{-1}(s) \ G^{-1}(ks)] \left[\frac{1}{s} \left(\frac{I_p}{I_p - F^{-1}(s)} \right) \frac{1}{ks} \left(\frac{I_p}{I_p - F^{-1}(ks)} \right) \right]^{-1} \tag{6}$$

Where:

k = 1,2,...,n.

n = 2 =PI Controller Structure

n = 3 =PID Controller Structure

In order to achieve a PI controller structure, the specifications is constrained for a particular process transfer functions as shown in equation (7).

$$F(s) = \text{diag}[F_1(s), \dots, F_p(s)] = \begin{bmatrix} \prod_{j=1}^{\hat{d}_1} \frac{a_{1ij}}{(s + a_{1ij})} & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & \prod_{j=1}^{\hat{d}_p} \frac{a_{pij}}{(s + a_{pij})} \end{bmatrix} \tag{7}$$

\hat{d}_i is defined to be the difference of order of s term between the denominator and the numerator of system transfer function. Also, to acquire the controller parameters, Franke Kruger and Knoop [3] substitute all the s-terms in the equation into a numerical value represented by α as shown in equation (8):

Where:

s = k α , k=1,2

$$[K_c \ K_i] = -[G^{-1}(\alpha) \ G^{-1}(2\alpha)] \left[\frac{1}{\alpha} \left(\frac{I_r}{I_r - F^{-1}(\alpha)} \right) \frac{1}{2\alpha} \left(\frac{I_r}{I_r - F^{-1}(2\alpha)} \right) \right]^{-1} \tag{8}$$

The selections of α are limited to a certain range of values according to the type of output required and also the type of process model involved. The range value for α and 'a' can be determined by comparing the standard normalized form with the user-defined specification forms of transfer function, F. For example, let $d_i = 2$, and consider the following:

$$F(s) = G(s)$$

$$\frac{a^2}{s^2 + 2\xi as + a^2} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (9)$$

Using the Settling time percentage and damping ratio equation:

$$T_{st\%} = \frac{K}{\xi a} \quad (10)$$

$$\text{Damping ratio} = \xi = \frac{\sqrt{\ln(\%overshoot)^2}}{\pi^2 + \ln(\%overshoot)^2} \quad (11)$$

For critical damped system ($\xi = 1$):

$$T_{st\%} = \frac{K}{a} \quad (12)$$

For underdamped system:

$$T_{st\%} = \frac{K}{\xi a} \quad (13)$$

Where:

$T_{st\%}$ =System settling time, $st\%$ = percentage of settling time, K =constant value; ω_n = natural frequency.

According to Franke, Knoop [3], the range of acceptable α value are:

$$\frac{4}{100T_s}, \dots, \frac{4}{10T_s} \leq \alpha \leq \frac{4}{T_s} \quad (14)$$

$$\therefore \alpha = \frac{4}{T_{4\%s}} = \frac{4}{1 \text{ sec}} = 4$$

Therefore, from Eq. 13, the value of 'a' determines the settling time. For critically damped solution a suitable value of the constant value in Eq 10 can be found by solving Eq 15 for a step input by noting the percentage of the final output.

$$\therefore f(t) = R(1 - (1 + at)e^{-at}) \quad (15)$$

Table 1. Normalized Transient Output Response for Critical Damped System

T	f(t)	% Final Value Reached (Approximate)
0	0	0%
1/a	(0.2642)R	26%
2/a	(0.5940)R	59%
3/a	(0.8009)R	80%
4/a	(0.9084)R	90%
5/a	(0.9596)R	96%
6/a	(0.9826)R	98%
7/a	(0.9927)R	99%

Table 1 illustrates the transient response to the step input and provides an approximate relationship between the response and the value of 'a'. Hence, settling time of 4% is:

$$T_{1\%s} = \frac{5}{a}, \quad (16)$$

$$\therefore a = \frac{5}{T_{4\%s}}$$

Table 2 shows the solution for the underdamped solution. Let ξ equals to 0.5912 (\approx 10% overshoot).

Thus:

$$f(t) = R \left(1 - \left(\frac{1}{1 - \xi^2 + j\xi\sqrt{1 - \xi^2}} \right) e^{-a\xi t} \cos\left(at\sqrt{1 - \xi^2}\right) \right) \quad (17)$$

Table 2. Normalized Transient Output Response for Underdamped System

T	f(t)	% Final Value Reached (Approximate)
0	0	0%
1/a	(0.7454)R	75%
2/a	(0.7879)R	79%
3/a	(1.1274)R	113%
4/a	(1.0936)R	110%
5/a	(1.0327)R	103%
6/a	(0.9964)R	100%
7/a	(0.9872)R	99%

Eq 6 represents the expression to determine the PI controller terms. Proportional and Proportional, Integral & Derivative Controller expressions can also be developed, if necessary, in a similar manner. Although this method allows the system to obtain controller parameters in terms of the standard PID controller, it may not achieve the desired specifications due to loop interactions that are a usual occurrence in many systems. A GA will improve on the objective functions values as it tries to optimise the system performance according to the specifications. All these equations are correct if the process used has the same number of inputs and outputs.

Example 1.0

The process used in this example is a 2-input, 2-output multivariable process. Consider the process state space Eq 7 that describe the process model [6]:

$$A = \text{diag}[-0.9321 - 0.9341 - 0.217 - 0.2159 - 11.59 - 8.057]$$

$$C = \begin{bmatrix} 0.68 & -1.6427 & 0.1252 & 0.2234 & 1.4194 & 0 \\ -0.0409 & 0.1558 & 0.0217 & 0.0646 & -1.5583 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Four Proportional & Integral Action (PI) controllers were implemented for this process according the following user specifications.

i) Critical damped system with 4% of system settling time of 1 second for both first and second outputs. Therefore:

$$T_{4\%} = 1s = \frac{5}{a} \quad \therefore a = 5$$

Thus according to the equations in (11):

a1st loop = 5

a2nd loop = 5

Using the Eq 7, the desired transfer function is as follow:

$$F_w(s) = \begin{bmatrix} \frac{5^2}{s^2 + 2(5s) + 5^2} & 0 \\ 0 & \frac{5^2}{s^2 + 2(5s) + 5^2} \end{bmatrix}$$

Consider equation (8), s-terms are then substituted with numerical value. The controller parameters are:

$$K_c = \begin{bmatrix} 1.4094 & 1.8358 \\ -13.6659 & 5.5431 \end{bmatrix} \quad K_i = \begin{bmatrix} 5.091 & 11.6515 \\ -31.723 & 17.4037 \end{bmatrix}$$

Applying a unit step input to each of the system inputs using MATLAB/SIMULINK [5] package. The system transients are as shown in Figure 6.

ii) Underdamped system with settling time of 4 seconds and 10% overshoot for both outputs. Therefore:

$$\xi = \sqrt{\frac{\ln(\%overshoot)^2}{\pi^2 + \ln(\%overshoot)^2}} = 0.5912$$

$$\omega_n = 1.6916$$

From Table 2, the 1% settling time is considered, thus:

$$T_{4\%s} = \frac{7}{\xi a}$$

$$\therefore a = 2.960, 1s \approx 0.57 * a$$

$$F(s) = \begin{bmatrix} \frac{3.0625}{s^2 + 1.179s + 3.0625} & 0 \\ 0 & \frac{3.0625}{s^2 + 1.179s + 3.0625} \end{bmatrix}$$

Thus, the controller parameters value:

$$K_c = \begin{bmatrix} 0.0951 & -0.5018 \\ -3.3037 & 0.2859 \end{bmatrix} \quad K_i = \begin{bmatrix} 1.7961 & 4.9595 \\ -8.4561 & 5.1785 \end{bmatrix}$$

Applying a unit step input to each of the system input pattern, the transient outputs in Figure 8.

3. Genetic Algorithms Optimization

The main title (on the first page) should begin 1 3/16 inches (7 picas) from the top edge of the page, centered, and in Times New Roman 14-point, boldface type. Capitalize the first letter of nouns, pronouns, verbs, adjectives, and adverbs; do not capitalize articles, coordinate conjunctions, or prepositions (unless the title begins with such a word). Please initially capitalize only the first word in other titles, including section titles and first, second, and third-order headings (for example, “Titles and headings” — as in these guidelines). Leave two blank lines after the title.

Genetic Algorithms (GAs) implement an optimization technique based on a simulation of the natural law of the evolution of species via natural selection, in order to have the fittest individual to survive. The searching process of GAs is similar to the natural evolution of biological creatures in which successive generations are created and raised and they themselves continue the cycle. In this algorithm, the fittest among a group of artificial species, which are represented in a form of string structure, survive and form a new generation with those that are produced through structured but random information. In every new generation, a new offspring or set of strings is created using information of the gene of the fittest old generation. This allows GA to exploit the historical information of a ‘species’ in order to have gradual improved characteristics or behaviour.

The GA can be mainly classified into three parts. First is the structure of strings, that includes a coding and decoding method, second is the fitness function that defines the specific performance requirement, and finally, the genetic operators which involve reproduction, crossover and mutation.

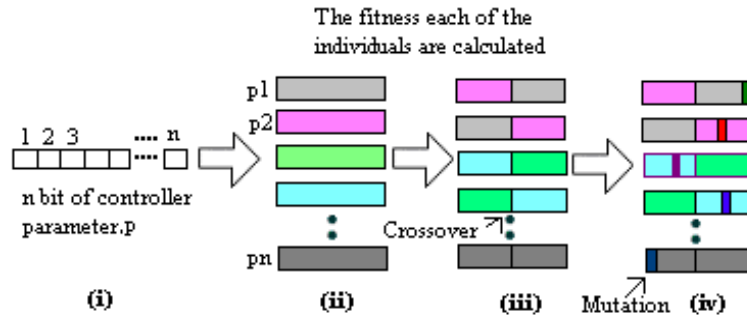


Figure 2. GA Optimisation Technique

Where:

- i) Coding of controller parameters in a binary string
- ii) Create number of individuals with different and random values
- iii) New strings are produced via crossover in pairs
- iv) Once the crossover is implemented, mutation is applied to each new string according to a mutation rate

The control input of a PI controller can be represented as below:

$$u = K_I \int e(t) dt + K_c(-y(t)) \quad (18)$$

Where K_c is the proportional gain, K_I is the integral gain and e is the error of system output. In order to represent the controller parameter in a GA, binary vectors are used. The initial population may be generated randomly. Alternatively, a member of the population can be defined, for example, by using a Z-N tuned controller, DSM or the revised Direct Synthesis Method.

A. Objective Functions

In a natural system, species will evolve adapting themselves to the environment. In GA terms, the controller parameters, *i.e.* the species, will try to adapt to the objective functions, the environment. The objective functions that may be used in tuning the PI controller parameters in a system are in the form of a specified performance of the system. The objective function is the difference between the system actual output and the desired system output, as shown in Figure3, when a unit step function applied to i th input, as equate in Eq. 19.

$$J_{ij} = \int_0^t (y_{ij}(t) - f_{ij}(t))^2 dt \quad j=1,2,..m \quad (19)$$

Each objective function are weighted and added. Therefore, the final objective function is:

$$J = \sum_{i=1}^n w_{ij} J_{ij}$$

Where:

- $y(t)$ = system output
- $f(t)$ = specified system output
- n = number of outputs
- m = number of different set points applied to system inputs
- w = weighting constant

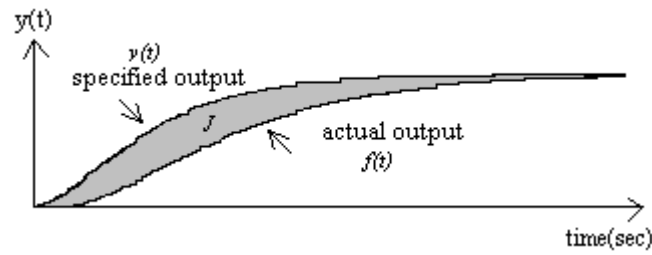


Figure 3. Error Signal

The aim is to minimize the value of J in order to obtain a good fitness value. However, since the GA is an optimization tool, the fitness function will be calculated as:

$$\text{Fitness, } F = \frac{1}{J} \quad (20)$$

After a certain number of generations, fitness of the dominant string will be higher than the previous generation. Eventually, an optimal string will be obtained that contains the highest fitness value. This technique has the advantage obtaining acceptable system performance according to user specifications and the final controller parameters are in the standard form of the PI controller configuration.

Example 2.0

i) Using the same specifications in Example 1.0, the chromosomes are subjected to single point crossover at every generation with generation gap, crossover and mutation probabilities of 0.9, 0.5 and 0.03 respectively, with maximum generation of 100. The weighting constants are set as $w11= w22=1$, $w12= w21=0.1$. The initial individuals of the population are randomly selected. The selection for creating new offspring attained with stochastic universal sampling [1]. The reinsertions of the chromosomes back to the population are based on their rank fitness calculation. Also, at each generation, the best parent chromosome will replace the worst performed offspring chromosome. Each controller parameters are encoded using 20 bits resulting in total of 160 bits in a single string. This is to ensure the accuracy of conversion of the chromosomes into controller parameter values. The system simulations are implemented using the Runge-Kutta fifth order numerical integration with a constant step size of 0.1 second.

Graph in Figure4 demonstrates that the objective function value converged less than 50 generation. The controller parameters obtained are:

$$K_c = \begin{bmatrix} 8.5584 & 3.6151 \\ -16.217 & 9.8889 \end{bmatrix} \quad K_i = \begin{bmatrix} 22.3517 & 14.6603 \\ -38.700 & 28.5651 \end{bmatrix}$$

ii) For the underdamped system problem similar GAs operators are applied. The controller parameters are:

$$K_c = \begin{bmatrix} 1.3858 & 0.5678 \\ -1.9231 & 1.8071 \end{bmatrix} \quad K_i = \begin{bmatrix} 6.0122 & 4.9719 \\ -1.8742 & 9.7241 \end{bmatrix}$$

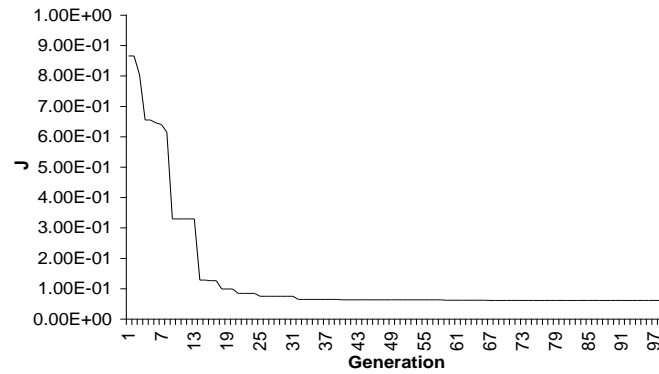


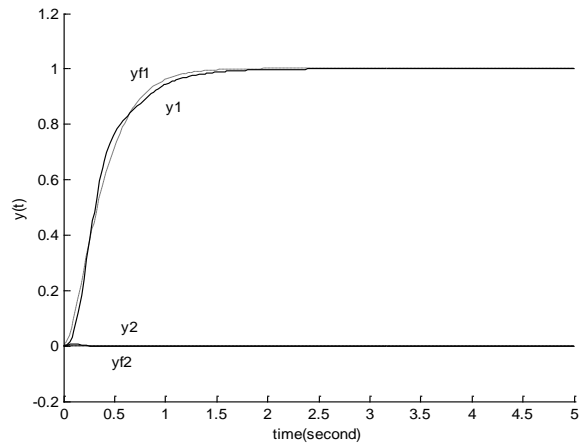
Figure 4. Best Objective Values for Critical Damped System

4. Discussion and Conclusions

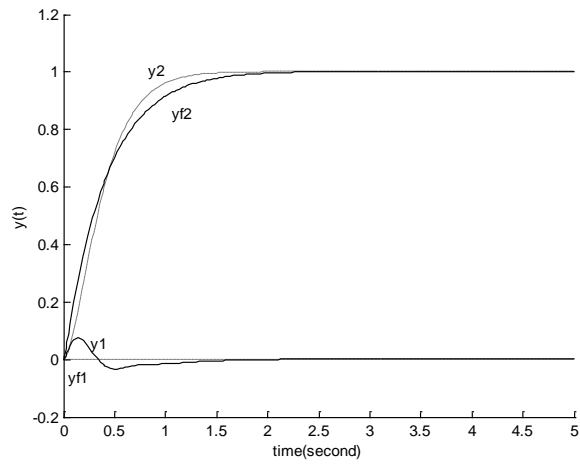
The outcomes of the system using PI controller to obtain the desired result in Example 1 using the mDSM, shows that the method achieved acceptable performance as illustrated in Figure 5 (a) & (b). However, when the system was required to attain a certain percentage of overshoot, the second output could not achieve the desired performance. Especially for the second loop output, the transient shoots up to almost 20% overshoot.

The GA was implemented to the same system and uses the same applications. The first exercise showed that GA managed to achieve the desired specifications and the controller parameters are in PI form. The best objective functions of each population converged in less than 50 generations. Also, the GA managed to produce a slightly better outcome of the second output than the system that uses the mDSM. While the second requirement in Example 2, GA performed better than the modified Direct Synthesis. The best objective function values for each generation converged as the 100 generations were reached.

In this paper, the comparisons were made between the mDSM and the GA method in tuning a decentralized PI for a multivariable system. The DSM works really well in a SISO system, the structure becomes complex in a MIMO system and the controller derived may not be realisable. The mDSM ensures a realisable and specific controller structure by substituting the s-terms in the analysis with a numerical value. The specifications transfer functions has to be chosen carefully to ensure the required controller form is achievable. GAs provides a flexible approach to controller design, for specification, controller structure and choice of cost function to achieve the objective. The differences between the desired output and the actual system outputs are used within the desired objective function to enable the GA to search for the best possible controller parameters.

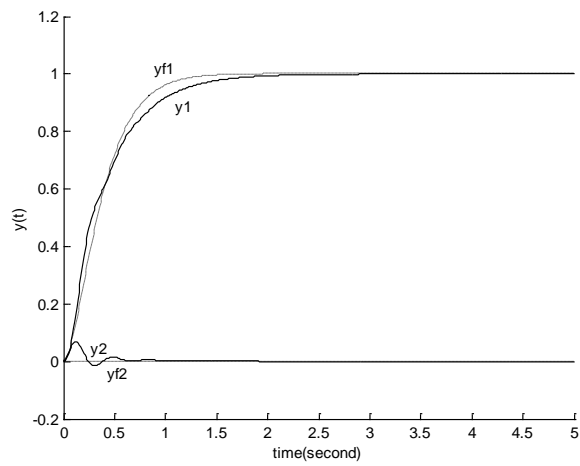


(a) Unit Step Signal as the First Input

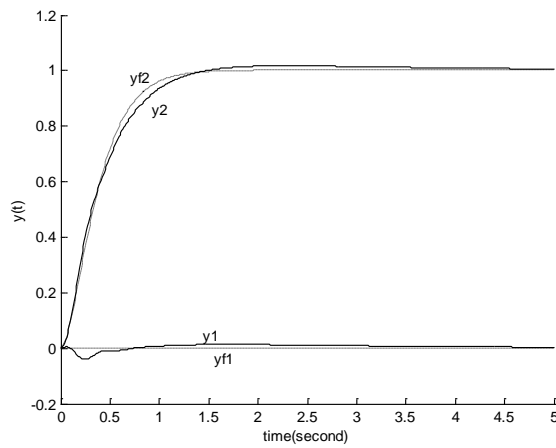


(b) Unit Step Signal as the Second Input

Figure. 5 Transient Response for Critical Damped Output



(a) Unit Step Signal as the First Input



(b) Unit Step Input as the Second Input

Figure 6. Transient Response for Critical Damped Output Using GAs

References

- [1] J. E. Baker, "Reducing Bias and Inefficiency in the Selection Algorithm", Proceedings of the 2nd International Conference on Genetic Algorithms, (1987); New Jersey.
- [2] K. A. De Jong, "An Analysis of the Behaviour of the Class of Genetic Adaptive Systems", PhD thesis, University of Michigan, (1975).
- [3] D. Franke, K. Kruger and M. Knoop, "System dynamik and Reglerentwurf", R. Oldenbourg Verlag Munchen Wien, (1992).
- [4] J. H. Holland, "Adaptation in Natural and Artificial Systems", University of Michigan Press, (1975).
- [5] MATLAB/SIMULINK Version 6.0, "The Language of Technical Computing", Release 12, (2000).
- [6] R. V. Patel and N. Munro, "Multivariable System Theory and Design", International Series on Systems and Control, Pergamon Press, vol. 4, (1982).
- [7] D. E. Seborg, T. F. Edgar and D. A. Mellichamp, "Process Dynamics and Control", 2nd ed. John Wiley and Sons, (1989); New York.
- [8] C. Vlachos, D. Williams and J. B. Gomm, "Solutions to the Shell Standard Control Problem Using Genetic Algorithms", Proceedings of the UKACC International Conference on Control, (1998); UK.
- [9] C. Vlachos, D. Williams and J. B. Gomm, "Genetic Approach to Decentralised PI Controller Tuning for Multivariable Processes", IEE Proc.-Control Theory, Appl, (1999).
- [10] P. Wang and D. P. Kwok, "Auto-Tuning of Classical PID Controllers Using An Advanced Genetic Algorithm", IEEE International Conference on Neural Network, (1992).
- [11] J. G. Zeigler and N. B. Nichols, "Optimum Settings for Automatic Controllers", Trans ASME, (1944).

Authors



Lifei Deng, is currently a Ph.D student at Jilin University, China. She received her master's degree at Jilin University in 2014. Her research interests are in the fields of signal processing, and engine control theory, system modeling.



Yaowu Shi, is currently a professor at Jilin University, he is director of the Jilin Institute of automation, senior member of the China Electronics Society, director of the China weak signal detection institute. His research interests are in the fields of signal processing, and engine control theory, system modeling and optimal control theory.



Lanxiang Zhu, is currently a lecture at Changchun Architecture and Civil engineering College, China. She received her Bachelor (1st class) degree from Jilin Jianzhu University, Master of Science degree from Harbin Institute of Technology, and PhD degree from Jilin University in 1988, 1998, and 2013 respectively. Her research interests are in the fields of signal processing, and engine control theory, system modeling and optimal control theory.



Dingli Yu, is currently a professor at Liverpool John Moore University, He received his Bachelor (1st class) degree from Haer Bin Jianzhu University, Master of Science degree from Jilin University, and PhD degree in 1982, 1986, and 1995 respectively. Dingli Yu is the member of the international automatic control (IFAC) fault detection and Technical Security Council, the assistant editor for International Journal pattern recognition and control.



Rui Zhu, is currently an employee at Changchun zonbo new energy transmission system of limited liability company. He received his Bachelor degree from Changchun University of Science and Technology, China. His research interests are in the fields of the mechanical design and its self - moving.

