An Alternative Structure to Replace the Tree-Based Individuals in a Genetic Programming Algorithm for Decoupling MIMO Systems

Martín Montes Rivera[1], Marco Antonio Paz Ramos[1], Jorge Luis Orozco Mora[2], Carlos Alberto Ochoa Zezzatti[3]

[1]Universidad Politécnica de Aguascalientes, Aguascalientes, Ags., México
{martin.montes, marco.paz}@upa.edu.mx
[2]Instituto Tecnológico de Aguascalientes, Aguascalientes, Ags., México
drorozco@mail.ita.mx
[3]Universidad Autónoma de Ciudad Juárez, Ciudad Juárez, Chi., México
alberto.ochoa@uacj.mx

Abstract. MIMO systems control is more complex than SISO systems control because its variables are coupled, but multivariable control techniques can get them decoupled. There are many MIMO systems in the industry but the majority are controlled in ranges where is possible using SISO systems techniques, because the difficulty of multivariable control. Genetic Programming is a technique that can be used obtaining mathematical models but the program design is complex and requires great computational power to get good results. In this paper an alternative algorithm using genetic programming is employed for decoupling a paper machine process and an irrigation system of fields, the complexity of genetic programming technique is simplified with a new propose equation structure that replace the classical tree-based individuals, the equation structure reduce the time creating individuals and evaluating its fitness value, using the new equation structure is possible to obtain decoupling blocks for decouple the MIMO systems.

Keywords. Genetic programming, multivariable control, decoupling of MIMO systems, mathematical models.

1 Introduction

The main objective of control theory is to achieve a desired behavior in controlled variables, and it could be the input tracking or system stabilization [1].

When systems have several inputs and outputs are denominated Multiple Input-Multiple Output (MIMO) and systems with only one input and one output are called Single Input-Single Output (SISO) [2-4].

Decoupled MIMO systems have several inputs and outputs, but there is not relationships among its variables, which means that they can be easily separated in one SISO system per every combination of input and output. When MIMO systems are coupled their inputs and outputs have relationships among its variables and to get them decoupled classical Multivariable Control (MC) techniques are required [5-7].

MC is important because there are many MIMO processes in the industry, but the majority of them are controlled in limited ranges using the SISO systems techniques, although systems remains uncoupled, this partial solution is used because classical MC techniques are more difficult compared with SISO systems control, [8-10].

The classical MC techniques are focused on the suppression of relationships among the variables of MIMO systems, decoupling blocks or gains, are commonly used to suppress this relationships, but complex mathematical models are needed to obtain the decoupling blocks and this does not guarantee results, i.e. with the mathematical model it is not always possible to get a decoupled system [4], [11].

An alternative technique to get decoupling blocks for coupled MIMO systems is to obtain them by using Genetic Programing (GP). GP is an evolutionary algorithm proposed by John Koza, this technique is able to develop computer programs by a stochastic form, which is related to natural selection studied by Charles Darwin [11-12].

Getting linear and nonlinear models using GP does not need human work during program execution, although the algorithm which obtains them needs to be developed [11]. In addition, the GP algorithm is able to test several cases until it finds the solution and suppress the complex analysis used in classical MC techniques, obtaining the decoupling blocks for coupled MIMO systems.

GP algorithm is capable of obtaining unknown mathematical functions or models, by using natural selection criteria during several generations on a population of possible solutions, but the program design is complex and after finished requires great computational power in order to get good results [11].

The population of a GP algorithm is commonly constructed by using a set of tree-based individuals that represent the possible program solutions. Tree-based representation involves several conversions during GP algorithm, between the tree structure and the program represented [12-14].

In this paper an alternative equation structure is proposed to replace the classical tree-based individuals in a GP population, this new structure reduce time creating individuals and calculating its fitness value, also simplifies the GP algorithm design.

The power of the equation structure, is tested on a GP algorithm, where difficult known mathematical functions are proposed to be found. The new structure is also tested in a generator of decoupling blocks using GP, this is used to decouple a representative plant of a paper production machine process and an analog irrigation system of fields represented by a Resistor-Capacitor (RC) circuit.

Paper transport machine representation was elected to test the new equation structure in a generator of decoupling blocks using GP athwart to a complex MIMO system and the coupled RC circuit was used in order to test a simpler scenario.

2 Concepts of Intelligent Computing and GP Algorithm Design

The recent improvements in computational power related to increase of memory and processor speed, has benefited the soft computing techniques including GP, this new computational power makes possible to use soft computing tools on real applications like described in [14-17].

2.1 Intelligent Computing and Evolutionary Algorithms

Intelligent computing solve problems, using computational algorithms that does not require human work during its execution and emulate the intelligence of natural organisms. Intelligent computing has techniques like neural networks, fuzzy logic and evolutionary algorithms among others [17].

Evolutionary algorithms are based on natural selection principles proposed by Charles Darwin in his work “The origin of species”, where organisms with better adaptations to their environment survive and reproduces to pass their genes to the next generation. The process of an evolutionary algorithm starts creating an initial population, then some individuals of the population are randomly selected to change their structure, after that, all individuals are tested against the problem and the best possible solutions are selected to pass their genes to the next generation, finally, the population is rearranged according to its fitness value, which is a number that quantifies the capabilities that an individual has to solve a given problem, this process is repeated by several generations until it finds the required solution [18].

2.2 GP Algorithm Techniques

GP belongs to the evolutionary algorithms and follows their algorithm process, the steps of this process are follow creating programs that has unknown structures and are only limited in their extension, this programs are generated as a possible solution for a given problem [11-13].

Individuals. The representation of individuals or possible solutions in a GP population are commonly using a tree-based structure. Tree-based structure is similar to a tree because has a root and leaves, leaves are integrated by operators, variables or constants and the root is the first operation like is shown in Fig 1, a tree based individual is only limited by its depth, which is the number of levels since the root until the last level of leaves (Fig 1) [12-14], [16-18].

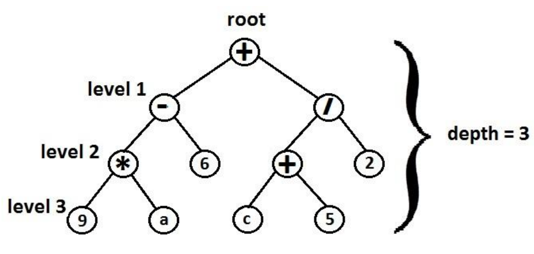

Fig. 1. Tree-based individual with depth of 3 levels.

Individuals Representation. The tree-based representation has a form that requires a representative word, so that the individual can be programmed reducing the required storage and pointers which reduces the efficiency of GP algorithm [13]. The representative word of the individual in Fig. 1 is shown in equation (1).

$$+(-(*(9,a),6),/(+(c,5),2)) \tag{1}$$

Population. Full and grown are common algorithms to create a GP population. Full algorithm creates all individuals with the same depth selected by user and grown algorithm creates individuals with any depth under a desired depth limit [12-13].

Fitness Value. The selection of individuals must be according to its fitness value, which is calculated by using the fitness function, this is an equation that compare the individuals with the desired characteristics to solve the problem. Fitness function can be the difference between an output and a desired output in control systems [12-13].

Tournament. The most common technique used in the selection of individuals is the tournament [13]. Tournament randomly selects sets of a defined number of individuals, then a winner is selected per set according to its fitness value and the individual winners performs a crossover operation to generate new elements [13-14].

Crossover. The mix of genes from the parents is made by a crossover operation, which first step is to randomly select a crossover point and then the tree-based parents are separated in sub-trees according to the place of the crossover point, finally their sub-trees are mixed to create new individuals, like in Fig. 2 [13-14], [16-18].

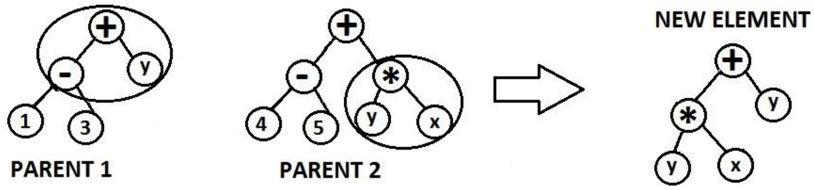

Fig. 2. Crossover operation between two parents generated with grown algorithm.

Mutation. This technique is an asexual reproduction that allows to include required genes that was not in the initial population. The process of mutation imply the random selection of an individual and this is mixed by crossover operation with a new created individual [12-14].

Elitism. The elitism maintain only the better adapted individuals for its reproduction and is programmed to avoid crossover operation of worst adapted individuals [12-13].

3 Program Design to Find a Proposed Function

Develop of a GP algorithm starts with the definition of initial values, this define behavior, desired solutions and limitations of the GP algorithm. The initial values are number of generations that algorithm is executed before stop, desired fitness value, individuals depth and their quantity in the GP population, after defined this values every technique is programmed following the section 2.2.

The creation of the GP program to find a proposed known function, requires the initial definition of the function and include it in the calculus of the fitness value. The fitness function must be programed to represent the difference between the evaluated value for coordinates x and y, using the proposed function and the individual that is being evaluated, like is in equation (2).

$$Fa = \sum_{x=0}^{a} \sum_{y=0}^{b} \left| f(x,y) - f_{ind}(x,y) \right| \tag{2}$$

Fitness function in equation (2) will generate a fitness value produced by the error between the known function and the individual, when the difference is minor the fitness value will be near to cero and the individual will have a better adaptability for the problem solution.

After program the algorithm to find proposed functions following GP process with tree-based individuals and using function aptitude in equation (2), was found that good results were produced but requiring great storage and time execution, which is related to great computational power demanded, like is mentioned in [11] and [14].

3.1 New Alternative Equation Structure

The analysis of GP algorithm to find proposed functions, shows that the main problem related to slow speed and great used storage, is due to the number of conversions required to represent an equation or mathematical function by using a tree representation. Conversions between a representative words of tree-based individuals and its represented equation, requires scanning for operators, variables and constants, then link operators to variables, producing the represented equation, so that fitness function be evaluated, like is shown in Fig. 3.

The use of conversions begins when the individual is created, in this moment the tree-based individual is generated by creating a representative word which contains the leaves in the tree representation, after generate the individual, fitness function must be evaluated and another conversion must be done, so that the representative word be expressed in an equation form to obtain the fitness value using equation (2), the tree representative word is used again in crossover operation and when the new individuals are obtained the conversion is used again to get their fitness value. The conversions increase time execution because are performed during several generations per every individual until the solution is obtained.

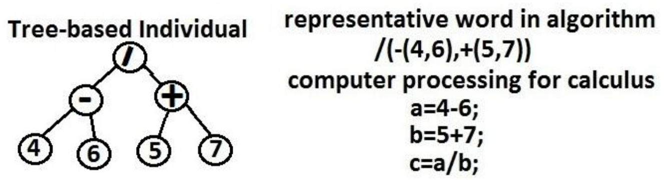

Fig. 3. Resume of the required conversions in a tree-based individual in the GP algorithm.

3.2 Equation-Based Individuals

The change of the individual representation by an equation, will simplify the process and also will reduce the execution time and storage required, in addition an equation representation is easily understandable by computer systems. Equation structure have operators, constants and variables, but without the required tree structure demanded in classical GP algorithms, the structure will not be limited by connections and is comprehend by any user because has the same structure that common equations, in Fig. 4 is shown a new equation-based individual.

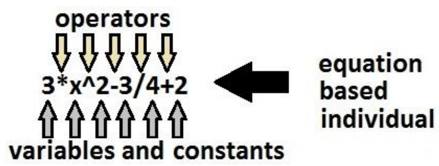

Fig. 4. Representation of new equation individuals.

Equation structures can contain any kind of operators like power represented by ^, which was not used in tree structures because operators from tree representations require at least two new leaves per operator. The use of new equation structure suppress the conversions used in tree-based individuals, because the equation-based individual has no representative word, the equation is the individual and does not require any conversion to evaluate the fitness function, like is shown in Fig. 5.

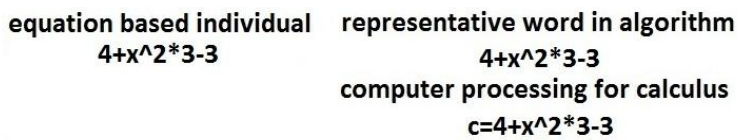

Fig. 5. Resume of GP algorithm for new equation-based individuals with no conversions.

The equation-based individual has problems with functions like described in equation (3), because this representation has no parenthesis to separate the operations in groups.

$$\frac{x+y}{x-y} \tag{3}$$

3.3 Equation-Based Individuals with Parenthesis

Equation-based individuals with parenthesis in operations without reduce the code speed or increase the storage, must be implemented so that any function be reachable. The equation-based individuals inside an array can represent parenthesis when operations are evaluated per element, and are read it like is shown in Fig. 6.

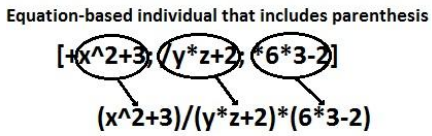

Fig. 6. Reading of an equation-based individual that includes parenthesis.

The main reason that explain the use of an array besides parenthesis, is to read easily the individual without scan by the parenthesis characters during GP algorithm. When array is used, the access per operation only requires the evaluation of the selected element and the operation can be done without any conversion. The first operator in every element is to connect the next operation. First element has an operator without use while evaluating the fitness value, but is required during crossover so that two new individuals can be created.

3.4 New Population Algorithm

Population algorithm for individuals with equation structure will be grown or full like in classical GP algorithm, but the limits are not defined according to depth, because there are no levels in equation-based individuals, the limits in this structures are the number of operators per element, and the number of elements per array.

3.5 New Crossover Operation

With equation structure inside of an array to represent parenthesis per element, a new scheme of crossover operation must be defined, because classical crossover operation requires concepts like levels and sub-trees, which does not exist in equation-based individuals.

Crossover operation of equation-based individuals is similar to the used in genetic algorithms, where a crossover point is selected and before of this point, the genes of a parent are used and after crossover point the genes of the other parent, if the genes of parents are arranged in inverted order to generate a new element, it is possible to create two new individuals per crossover operation like is shown in Fig. 7.

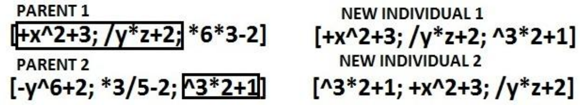

Fig. 7. New crossover operation using equation-based individuals.

3.6 New Mutation and Elitism Techniques

Mutation. This technique use the same concept described in section 2.2, and the only change is in how crossover operation is done, because the equation structure requires a crossover operation according to section 3.5.

Elitism. The elitism technique uses same concept described in 2.2, this is present in tournament selection where only better adapted individuals per tournament are reproduced and in a programed function to delete individuals with the worst fitness bellow a desired number of admitted individuals, contributing to maintain ecologic stability.

4 MIMO coupled Systems

The paper production machine process is a coupled MIMO system like irrigation of fields process, i.e. when one input is changed more than one output is affected, after decoupled this MIMO systems could be separated in SISO systems and controlled with SISO techniques, but with greater ranges without the limitations of coupled systems.

4.1 Paper Production Machine

Paper industry involves MC, because paper transport is a continuous production line that required the control of speed and tension in paper, which are coupled variables, because while the speed of paper is changed, tension must be controlled to avoid the paper rupture and when tension is changed speed of paper transport is affected.

An analog plant of a paper machine is the coupled transmission device CE108 developed by TecQuipment^{TM}, this apparatus uses a belt that represents paper and has three pulleys, two are linked to motors that produce the movement in paper transport and a third upper pulley allow measurement of speed and tension in belt.

4.2 Irrigation of Fields System

The irrigation process involves the use of water pumps for take water from wells to irrigated fields and valves are used to control water flow while irrigating.

Representing the irrigation plant with an analog RC circuit, power supplies in circuit imitate water pumps, resistors act as water valves and capacitors reproduce irrigated fields, like is shown in Fig. 8.

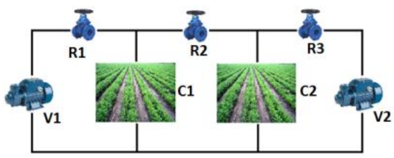

Fig. 8. Comparison between irrigation of fields system and RC circuit.

Irrigation of fields system has two coupled inputs represented by the power supplies, when a power supply is closed the water on fields is reduced, and if the other is closed the same occurs to all fields, in a desired decoupled system the water per field is changed only by one power supply or water pump.

5 Results

First test of equation structure in GP, is to reach a desired function proposed or at least approximated with an admissible error. Second test was apply the GP algorithm with an equation structure in the obtaining of decoupling blocks for real systems.

5.1 Reachable Proposed Functions

The function test using equation structure in GP algorithm, employ the fitness function in equation (2) and the initial values: 2000 generations, 250 individuals of population size, 6 operations per individual, 30 individuals mutated per generation, 5 individuals per tournament, and 240 individuals admitted according to their fitness value.

Equations (4) and (5) was the proposed tested functions.

$$9\sin(6\mathrm{x}+\mathrm{y})+y \tag{4}$$

$$e^{-x}-y+x^2+\frac{e^{-y}}{3} \tag{5}$$

The first equation results are shown in graph A in Fig. 9, where the total error between proposed equation and better adapted individual is 144.5667 units, which means an average error per point of 1.445667 units.

Despite this error, when the functions are plotted a minimal difference is observable between the two functions, which is logical considering the complexity of the equation since there are trigonometric functions with variables inside them, however a better approximation can be calculated by using the structure obtained in a genetic algorithm.

The tool that helps to reach any trigonometric function is trigonometric interpolation, where the only needed functions are sine and cosine in a polynomial expression. Using trigonometric identities and trigonometric interpolation is possible to reach any trigonometric function with an error depending of the polynomial elements [19].

Results of second equation are shown in graph B in Fig. 9, where total error is 5.6316 and average error per point is 0.056316, a minimal error considering that the maxim value of function is around 100.3334 and the error once plotted is not visible.

The individuals obtained for first and second test are in equations (6) and (7).

$$\cos(\mathrm{x})\sin(\mathrm{y})-5\cos(\mathrm{y})-\mathrm{y}+7\sin(\mathrm{y})-\mathrm{x}^2 \tag{6}$$

$$x^2-0.0928e^{-x}-y \tag{7}$$

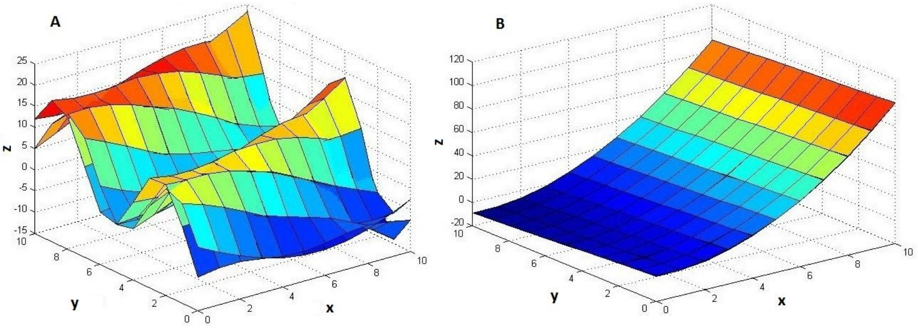

Fig. 9. Proposed functions approximated by using GP algorithm with equation structure.

After test the equation structure in the GP algorithm, the required time using the initial values was: 2.6538 seconds creating the initial population of 250 individuals with 6 operations, 4.3462 seconds evaluating the fitness value of this 250 individuals, an average time of 0.8112 seconds per generation and a total time of 1464.1 seconds since the algorithms starts until stops on the 2000 generation.

5.2 Decoupling of MIMO Systems

Initial values in the GP algorithm with equation structure for decouple MIMO systems where set: 4000 generations, 300 individuals of population size, 4 operations per individual, 30 individuals mutated per generation, 5 individuals per tournament and 240 individuals admitted according to its fitness value.

The obtained blocks with the equation structure in a GP algorithm for decoupling MIMO systems, must be connected summing in the system inputs so that represent the right links between inputs for decouple the system, like is shown in Fig. 10.

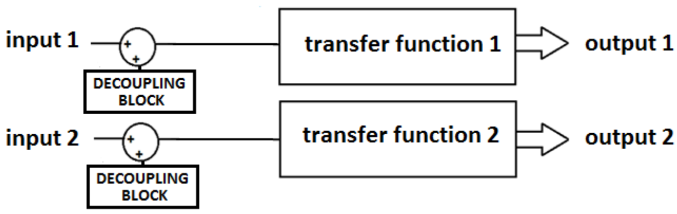

Fig. 10. Connection between system and decoupling blocks in a GP algorithm.

Fitness value for decoupling MIMO system must be calculated using the equation (8), this fitness function evaluate the blocks by measuring the output change when an input is changed, RC circuit decoupling blocks were obtained using fitness function in equation (8), but in paper machine process the equation (9) must be used, where a is the input 1 or 2 depending of which input is increased while block is evaluated, this change of fitness function is because the settling time of paper process was around 50 seconds, which makes the measurement process to slow evaluating fitness of decoupling blocks, but using fitness function in equation (9), the decoupling blocks can be evaluated in a desired range and this reduce time identification.

$$\sum_{input_1=0}^{10}\sum_{input_2=0}^{10}\left|output(\mathrm{x,y})-\mathrm{output(x-1,y-1)}\right| \tag{8}$$

$$\sum_{a=0}^{10}\left|output(\mathrm{a})-output(\mathrm{a}-1)\right| \tag{9}$$

Paper process is originally coupled since the top value of speed belt in coordinates (voltage motor 1=10, voltage motor2=10, speed belt=10) of graph A in Fig. 11 is reached by the use of the two motors, i.e. speed belt depends of motor 1 and 2, tension belt is increased depending of the difference between the voltages of motors, like is shown in graph B of Fig. 11, i.e. paper tension is also coupled and this behavior could produce paper rupture because when speed is increased tension also change its value.

The paper machine process can be decoupled if the obtained blocks using the equation structure in a GP algorithm 1.25x and –y are connected like is shown in Fig. 10, this blocks solve the problem because in graph C of Fig 11 the speed belt can be almost linear increased until the top value in coordinates (voltage motor 1=9, voltage motor 2=0, speed belt=10), this linear increasing is possible for ranges below of 2V in motor 2, which is used to control tension belt, when voltage in motor 2 is selected for greater values of tension, the system speed is affected by its tension, i.e. the system is decoupled for values of tension below of 2V in motor 2, however this works in a paper production process, because only requires a minimal tension while speed is controlled in different values.

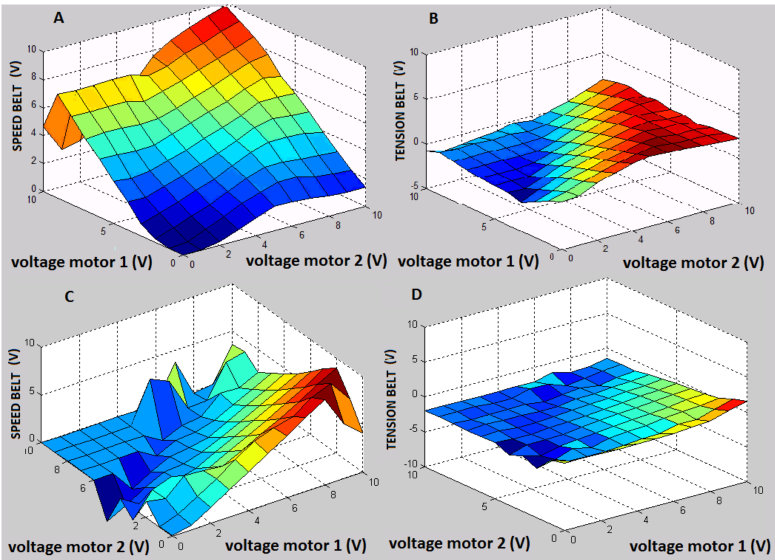

Fig. 11. Paper machine process coupled and decoupled with equation structure in GP.

The Irrigation process represented by the RC circuit is coupled because the greater voltages of capacitor 1 and capacitor 2 in coordinates (voltage 1=10, voltage 2=10, capacitor 1 voltage=10) and (voltage 1=10, voltage 2=10, capacitor 2 voltage=10) of graphs A and B in Fig. 12, are reached when the maxim voltages in sources are used and if one of the input voltages is changed the capacitor voltage change, i.e. capacitor voltages depends of the two input voltages in power supplies.

Irrigation process can be fully decoupled if the obtained blocks with an equation structure in a GP algorithms -0.5x and -0.5y are connected like is shown in Fig. 10, this blocks decoupled the system because every capacitor voltage can be controlled with one input voltage and the other input does not affect it, i.e. the maxim value for capacitor 1 is in coordinates (voltage 1=10, voltage 2=0, capacitor 1 voltage=10) of graph C in Fig. 12 and is maintained for every capacitor 2 voltage, and in capacitor 2 the maxim value is in coordinates (voltage 1=0, voltage 2=10, capacitor 2 voltage=10) of graph D in Fig. 12 and is maintained for every capacitor 1 voltage.

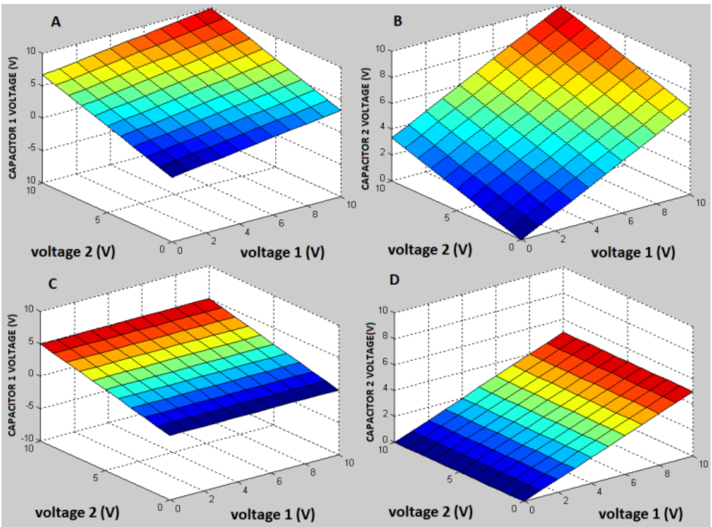

Fig. 12. Irrigation system coupled and decoupled with equation structure in GP.

Time for decoupling systems with a GP algorithm depends of the setting time in processes, in paper machine is 50 seconds while in RC circuit is of 0.8 seconds.

6 Conclusions

The tree-based individuals can be replaced by an equation structure in the obtaining of mathematical functions or models, since all proposed functions where reachable with a minimal error and the practical application in the obtaining of decoupling blocks for MIMO systems is successful, since the paper speed in paper machine process can be controlled while tension is selected below 2 volts and the irrigation of fields is decoupled because can be done with separated water pumps for each field.

Equation structures for individuals reduce time execution and simplifies the GP algorithm, because eliminate the required conversions between equation and tree representation individuals, when the classical tree based representation is used.

References

1. B. C. Kuo, Sistemas de Control Automático, Nueva York: Prentice Hall, 1996.
2. J. J. Espinosa Oviedo, Control Lineal de Sistemas Multivariables, Medellin, Colombia, 2003.
3. K. Ogata, Ingeniería de Control Moderna, Madrid: Pearson Educación, 2010.
4. F. Morilla, J. Garrido y F. Vázquez, "Control Multivariable por Desacoplo", Revista Iberoamericana de Automática e Informática industrial, vol. 10, pp. 3-17, 2013.
5. P. L. Falb y W. A. Wolovich, "Decoupling in the Design and Synthesis of Multivariable Control Systems", IEEE Transactions on Automatic Control, pp. Vol. AC-12, No. 6, 1967.
6. J. A. Delgado, J. A. Jáuregui, E. Rámirez V., S. E. Luna y J. L. Orozco, "Aplicación del Desacoplamiento Regular a un Circuito Eléctrico", Ropec Internacional, 2011.
7. L. Wang, Y. Cheng y J. Hu, "A Quasi-ARX Model for Multivariable Decoupling Control of Nonlinear MIMO System", Mathematical Problems in Engineering, vol. 2012, p. 13, 2012.
8. J. Acedo Sánchez, Control Avanzado de Procesos, Madrid, España: Ediciones Díaz de Santos, 2003.
9. A. Preglej, I. Steiner y S. Blazic, "Multivariable Predictive Functional Control of an Autoclave", Jounal of Mechanical Engineering, vol. 59, nº 2, pp. 89-96, 2013.
10. E. Eitelberg, "Load sharing in a multivariable temperature control system", Pergamon Control Engineering Practice, pp. Vol. 7 pp. 1369-1377, 1999.
11. B. Grosman y D. R. Lewin, "Automated nonlinear model predictive control using genetic programming", Computers and Chemical Engineering, pp. Volume 26 pp. 631-640, 2002.
12. J. R. Koza, Genetic Programming On the Programming of Computers by Means of Natural Selection, Cambridge, Massachusetts: Massachusetts Institute of Technology, 1998.
13. R. Poli, W. B. Langdon, N. F. McPhee y J. R. Koza, "A Field Guide to Genetic Programming", March 2008. [En línea]. Available: http://www.gp-field-guide.org.uk/. [Último acceso: 7 junio 2013].
14. W. P. Worzel, J. Yu, A. A. Almal y A. M. Chinnaiyan, "Applications of genetic programming in cancer research", The Internacional Journal of Biochemistry & Cell Biology, vol. 41, pp. 405-413, 2008.
15. M. Mahfouf, C. S. Nunes, D. A. Likens y J. E. Peacock, "Modelling and multivariable control in anesthesia using neural-fuzzy paradigms", Artificial Intelligence in Medicine, vol. 35, pp. 207-213, 2005.
16. C. Arturo, M. C. López, I. R. Andalon Garcia y M. E. Meda Campaña, "Genetic Programming as Alternative for Predicting Development Effort of Individual Software Projects", PLoS ONE, vol. 7, nº 11, pp. 1-10, 2012
17. H.-C. Tsai, "Using weighted genetic programming to program squat wall strengths and tune associated formulas", Engineering Applications of Artificial Intelligence , vol. 24, pp. 526-533, 2011.
18. D. B. Fogel, "Foundations of evolutionary computation", Proceedings of the SPIE, vol. 6228, 2006.
19. J. L. Buchanan y P. R. Turner, Numerical Methods and Analysis, Singapore: McGraw-Hill, 1992