Journal of Theoretical and Applied Information Technology
15th September 2020. Vol.98. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ADAPTIVE LEARNING SCHEME FOR THE VIRTUALIZATION OF A ROTARY SERVO BASE UNIT

[1]HOLMAN MONTIEL A., [2]FREDY H. MARTÍNEZ S., [3]EDWAR JACINTO G.

[1, 2, 3] Universidad Distrital Francisco José de Caldas, Facultad Tecnológica, Bogotá D.C., Colombia

E-mail: [1]hmontiela@udistrital.edu.co, [2]fhmartinezs@udistrital.edu.co, [3]ejacintog@udistrital.edu.co

ABSTRACT

A system is a structure composed of mechanical, electrical or electromechanical parts that interact with each other to fulfill an objective. At an industrial level they are known as manufacturing processes and at an academic level these processes are emulated by implementing mechanisms such as: designing scale prototypes, building test and trial laboratories or developing specialized simulators. However, the efforts made by the authors to build scale or simulated prototypes that are an ideal copy of the real process are not perfect. On the one hand, some real physical implementations reduce the margin of error by improving the quality of the prototype materials, but as the quality of the materials increases, so does their cost, which reduces accessibility to the population. On the other hand, the simulators do not perfectly emulate characteristics of the environment, such as humidity, temperature, vibrations, among others, which reduces its reliability in the presentation of the results obtained. Therefore, this article proposes a strategy to virtualize a QUANSER SRV-02 rotary servo base unit, which from experimental data reconstructs a mathematical model using a Genetic Algorithm (GA), which minimizes the margin of error between experimental and practical data. This tool will allow virtual practices (simulation) with results very close to the behavior of the real plant.

Keywords: *Genetic Algorithm; ARIMA Models; Servomechanisms; Virtualization; Prototypes.*
.

1. INTRODUCTION

The manufacturing processes in the industry are sets of multiple electrical, mechanical or electromechanical elements, which are used to manufacture some product of massive consumption or unipersonal. Sometimes, these processes are modified to improve the finished product, reduce manufacturing time or increase production capacity, which has led different manufacturers to develop techniques to parameterize manufacturing process lines and thus partially or totally estimate an expected result [1-4].

One technique for parameterization is based on building scale replicas, which represent each component of the process and the variables involved during its operation. For example, minimalist, modular and flexible manufacturing systems distributed by certain manufacturers, which allow the user to have a practical experience and to know a production line partially or totally [3-4].

Other techniques are based on the development of simulators with the help of programming environments with graphic environments, such as; flight or driving simulators. The simulators try to represent many characteristics of the real environment and give the user the experience to understand the functioning of the real process. Undoubtedly, there is many simulators and they are classified according to the area of knowledge, which indicates that a software-based application is easier to massify than a real physical implementation [5-6].

The massification of industrial-type simulators has allowed industrial designers to innovate in manufacturing processes, with relatively few physical resources compared to actual implementation [7]. Although, simulators are versatile and portable by increasing the amount of process variables to be simulated, so does the computational cost required by the application to provide results to the user. This is because a simulator in its simplest structure is a program that solves systems of equations, whose convergence

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

time varies according to the number of variables that each equation has [8].

Similarly, as simulators diversify, so do the techniques for solving systems of equations. Among the most widely used solution methods are differential equation calculators, which are basically applications that implement approximate methods to estimate a solution (commonly based on Runge Kutta, Euler, Monte Carlo methods). An advantage of these methods is that they are computationally simple to implement and provide a response very close to that given by the analytical method. Some variants to these solution methods are ARIMA (Autoregressive Integrated Moving Average) models, which are stochastic and statistical models used to find patterns in data sets, or computational learning approaches that rely on neural networks or vector support machines to represent data sets and estimate system response [9-11].

In summary, there are many ways to represent a manufacturing process, among them the physical implementation requires a higher monetary investment compared to a simulator [1,4,10]. However, the simulators have a margin of error with respect to the real process, because the simulator does not consider all the variables of the environment, such as the environmental conditions, the way in which the equipment is used or the amount of maintenance that is done to the process. Therefore, this article proposes a strategy to virtualize a system based on experimental data, in order to reduce the margin of error between the expected values and those provided by the model. Basically, the model is a mathematical expression that updates its coefficients with an optimization algorithm to fit a set of data provided by the user. This model and the results obtained are described in the following sections, which are organized in the following way: in numeral 2 a conceptual description is made before exposing the developed model, in numeral 3 the proposed model is presented and its operation is described. Finally, section 4 shows the results obtained.

2. MATERIALS AND METHODS

The model presented in this paper is obtained by modifying several coefficients of an equation using an optimization algorithm and the validation is done using an experimental model. The concepts used to develop the model are described in detail in this section.

2.1 Optimization Algorithm

The concept "optimization" is used in several areas of knowledge, since it is a process by which the best available value is selected within a group of values. In some cases, these processes are systematized by means of routines developed in computers, which assign different values to the input variables of a mathematical function and thus estimate their output value. Usually, these routines are converted into sets of methods or algorithms to solve optimization problems in an iterative way [11-15].

In this way, an optimization algorithm searches for solutions in two different ways. The first case is the minimization of the function where it looks for the minimum available value, which becomes the optimal value. The second case is the maximization where the optimal value is the maximum available value. It is worth saying that, in both cases, there are local or global optimal (see Figure 1), where a local optimal $F(x_1)$ exists if there is a $F(x_2)$ and it happens that $F(x_2) \leq F(x_1)$ o $F(x_2) \geq F(x_1)$ [11-13].

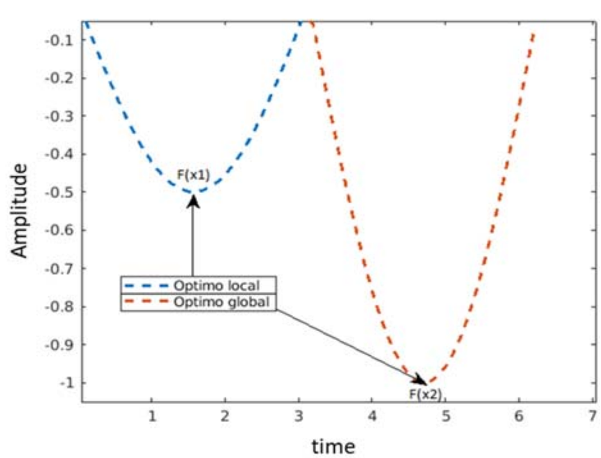


Figure 1: Example of local and global optimal value

Optimization algorithms have a finite number of iterations and try to find a rough solution to the optimization problem, for example, the Simplex Algorithm (SA) that solves linear programming problems [11]. Like the AS there are many optimization algorithms and ways to classify them, but in this article, we will deal with the Comprehensive Search Method (CS), Simulated Temperate (ST) and a Genetic Algorithm (GA).

CS's method estimates a set of candidate solutions to the problem, tests each solution in the function to be optimized, and among them selects the best [13]. However, CS's method can take a long time to arrive at a solution, so algorithms such as ST or GA are proposed.

The ST algorithm (see Algorithm 1) emulates the behavior of the heating and cooling process of the metal in a metallurgical process and is based on the generation of random values or candidate solutions whose selection criteria varies according to the result given when solving an exponential function. This exponential function is affected by a variable that allows its output value to be damped as the iterations of the algorithm increase [14].

Algorithm 1. Simulated Tempering

$\tau = 1$
Termination=0
While (¡Termination) **do**
    $G(\Delta\tau, \tau) \leftarrow f(x_1) - f(x_2)$
    $q \sim U[0,1]$
    If ($f(x_1) < f(x_2)$) o ($q < G(\Delta\tau, \tau)$) **then**
        $x_1 \leftarrow x_2$
    else
        $x_2 \sim U[-10,10]$
        $\tau++$
    If $\tau > 50$ **then**
        $\tau = 1$
        $G(\Delta\tau, \tau) = e^{-\frac{\Delta\tau}{\tau}}$

As you can see, CS and ST methods are search algorithms that update a single output value per iteration, and this can take a long time for a solution to be found. Therefore, population algorithms arise that implement search methods that in one iteration estimate multiple candidate solutions, such as GA.

A GA is an algorithm that emulates Darwinian natural selection, that is, each iteration (generation) updates a population of individuals by trying to make them potentially better. In this order of ideas, each candidate solution is an individual, a set of individuals is a population, and their behavior when trying to solve the problem is known as aptitude value [15-17].

The GA updates a population every generation according to a method of selection, crossing and mutation. The selection method is a function that generates a new set of individuals by discarding those with the worst fitness value. The crossing method takes elements of the created set of solutions, pairs them (Parents) and generates another set of solutions (Children). The mutation method changes characteristics of one or more individuals from the solution set (the Sons) [15-19].

2.2 Modelling

A model is a representation of a system, which on a scientific level establishes a relationship between a phenomenon and an abstraction of it. This type of abstractions at an engineering level are presented by means of mathematical expressions [9-11]. In this case, the test system is the QUANSER SRV-02 plant that is composed of a motor, a gear train and a rotational encoder that allows the speed to be measured (see Figure 2). In addition, this system has a model to carry out simulations based on equation 1 [20]. Where N2 is the spring gear, N1 is the gear coupled to the motor, J is the motor inertia, R is the motor winding resistance, Kf and Kt are the proportionality constants provided by the manufacturer.

$$H(s) = \frac{k_t}{JLS^3 + (JR + Lb)s^2 + (Rb + k_f k_t)s} \quad (1)$$

Figure 2: QUANSER SRV-02

It should be noted that, models can also be represented with experimental data regressions, which reconstruct an equation between measured and experimental data. For example, linear regressions [11]. In this case, the proposed algorithm selects the number of terms and coefficient values to reproduce the experimental data with a low margin of error, this description and the details of implementation are described in detail in the following section.

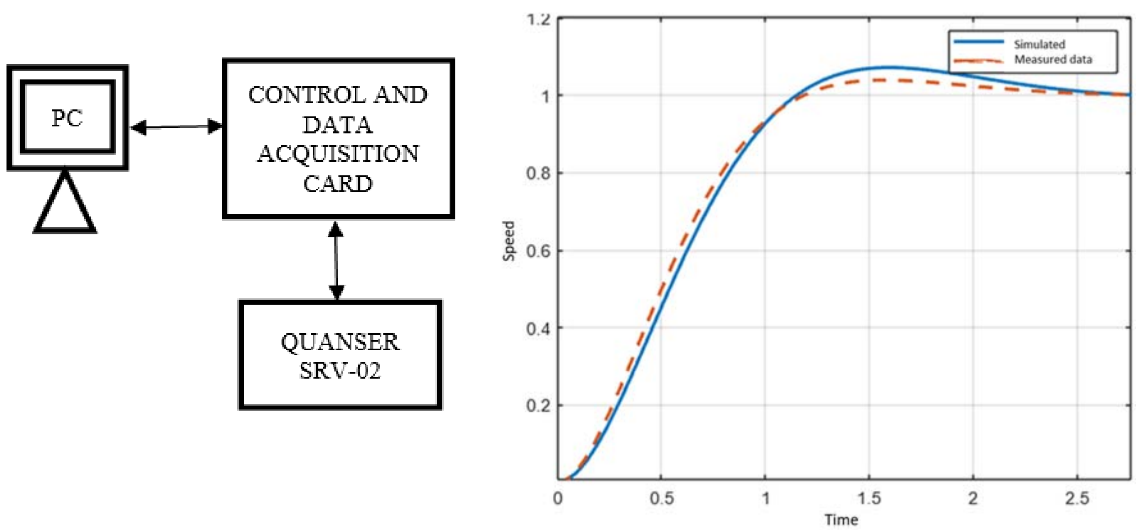
3. IMPLEMENTATION

The experimental data used to feed the model were obtained by switching on the QUANSER SRV-02 plant and connecting a data acquisition card that exports the data in a plain text file, see Figure 3 (a). There is a total of 250 records that show the speed against time and under the same conditions they were compared with the data provided by the available model to perform the system simulations, see Figure 3 (b).

Once the file was built, the optimization algorithms mentioned in the previous section were implemented. Considering that; the proficiency value is the average error between the experimental data and the model generated by the algorithm. An individual is the number of derivatives (maximum order 5) and their respective constant coefficient that are used to expand the polynomial of equation 2.

$$H(s) = \frac{AS^{n} + BS^{n-1} + \cdots CS^{0}}{DS^{n} + ES^{n-1} + \cdots FS^{0}} \tag{2}$$

The output of the system depends on a unit step function, that is, each time equation 2 is updated a step function is applied and compared with the analytical model to estimate the margin of error (see Algorithm 2). In the case of the CS and ST algorithms, which are search algorithms that use a single individual, the same number of evaluations were used as those used in the execution of the GA, in other words, a population of the GA that has 10 individuals is equivalent to 10 executions of the CS or ST.



a) Wiring diagram of the SRV-02 *b) Simulated and measured data*

Figure 3: Behavior of the QUANSER SRV-02 plant where the data obtained from measurements and simulations are observed.

Algorithm 2. How to generate an individual and estimate his aptitude value

Function Generate Individual ()
$Denominator\ [5];$
$Numerator[5];$
$D = Amount\ of\ denominator\ coefficients$
$N \sim U[0,5]$
$N \leftarrow round(N)$
$D \sim U[0,5]$
$D \leftarrow round(D)$
For i=0 **less than** N **do**
$Numerator[i] \sim U[-1,1]$
For j=0 **less than** D **do**
$Denominator[j] \sim U[-1,1]$
$H = Build\ function\ (Numerator, Denominator)$
Return H

Function Suitability value (**H**)
$y_1 = Apply\ unit\ step\ to(H);$
$y_2 = Apply\ unit\ step\ to\ the\ default\ model$
$error\_absolute = |y_1 - y_2|$
$error\_average\ =\ average\ (error\_absolute);$
Return error_average

Each optimization algorithm was parameterized differently to perform the same amount of proficiency value evaluations. Initially, CS's algorithm generates a set of 250 individuals at random and evaluates each of them, to select the best one. Then, the ST algorithm performs 250 evaluations where everyone is updated as shown in Algorithm 1. Unlike the other two algorithms, the GA generates a population of 10 individuals and runs for 25 generations. In addition, it is a modified algorithm to suit the situation, whose implementation is presented in Algorithm 3. Finally, the results obtained when implementing the different optimization algorithms in a computer (WINDOWS 10 operating system, an Intel inside TM core i3 processor, 8Gbs of RAM, a 240Gbs hard disk) and its behavior are presented in the following section.

Algorithm 3. Genetic Algorithm

Function Genetic algorithm ()
$P0[10] = Initial\ population;$
Generations=0;
$\mathrm{P0} \leftarrow$ Generate 10 individuals;
While Generations<25 **then**
$\mathrm{f} \leftarrow \mathrm{Evaluate\ (P0)}$
$\mathrm{P1} \leftarrow \mathrm{Selection\ (P0, f)}$
$P1 \leftarrow Crossing(P1)$
$P1 \leftarrow Mutation\ (P1)$
$P0 \leftarrow P1$
Generations++;

Function Selection (**P**, **f**)
$f\ = Normalize\ (f);$

$f, P \leftarrow$ Order from highest to lowest(f, P);
$P[7,9] \leftarrow P[0,2]$;
Return P

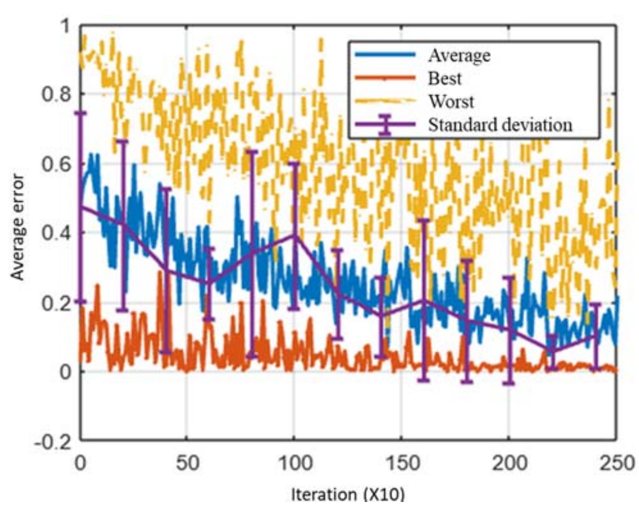
Function Crossing (**P**)
If $U(0,1) < 0.7$ **then**
$i_1, i_2 = Select\ two\ individuals\ at\ random(P)$;
If $U(0,1) < 0.5$ **then**
$i_1 \leftarrow numerator\ of\ i_2$
$i_2 \leftarrow$ numerator of i_1
else
$i_1 \leftarrow denominator\ of\ i_2$
$i_2 \leftarrow$ denominator of i_1
Return P

Function Mutation (**P**)
If $U(0,1) < 0.3$ **then**
i = Select an individual at random(P);
P = Generate a new individual and change the selected one(P, i);
Return P

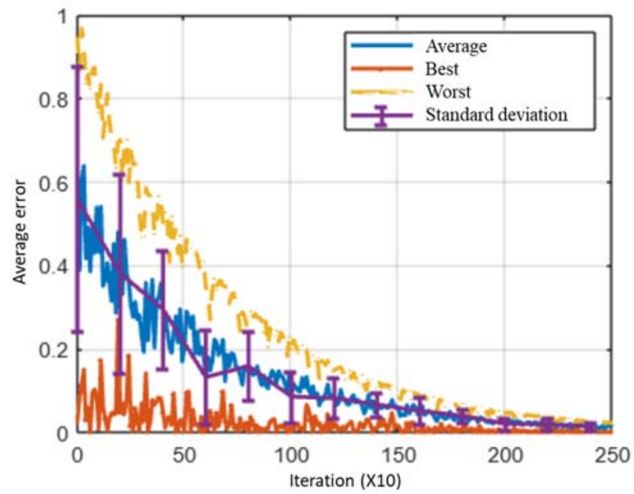
4. RESULTS AND DISCUSSION

Each algorithm mentioned in the previous section (see Figure 4) made 250 assessments of the proficiency value and this process was executed 10 times, with this information the standard deviation, the average, the best and the worst individual were estimated. In the case of the GA, two graphs were taken, one showing the behavior of all individuals individually (see Figure 4c) and the second graph showing the behavior of all individuals in a generational way (see Figure 4d).

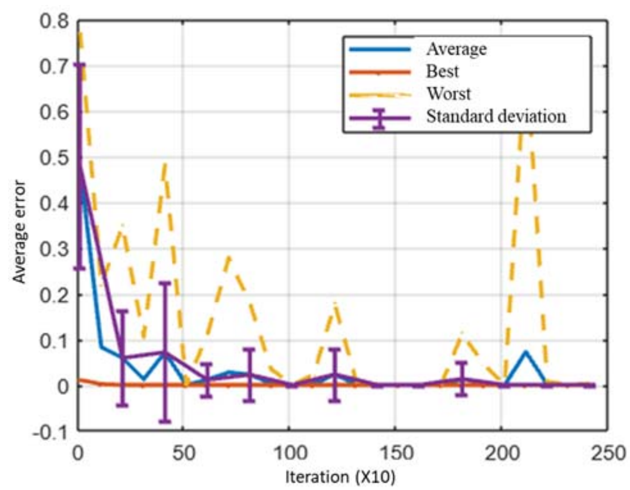
Of all the individuals evaluated, the best of the best was taken, that is, the best individual found in the 2500 aptitude value evaluations made by each optimization algorithm, whose graph is contrasted with the measured data (see Figure 5a) and the simulation response provided by the manufacturer (see Figure 5b). Finally, the trend of the graphs presented, and their interpretation is made in the following section.



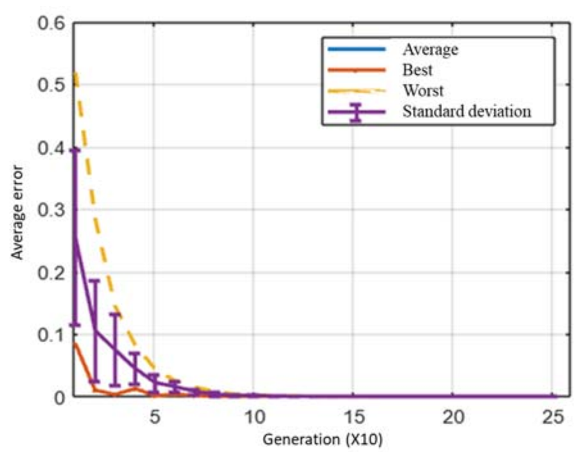
a) Comprehensive Search (CS)



b) Simulated Temperate (ST)



c) Genetic Algorithm (overlapping individuals)

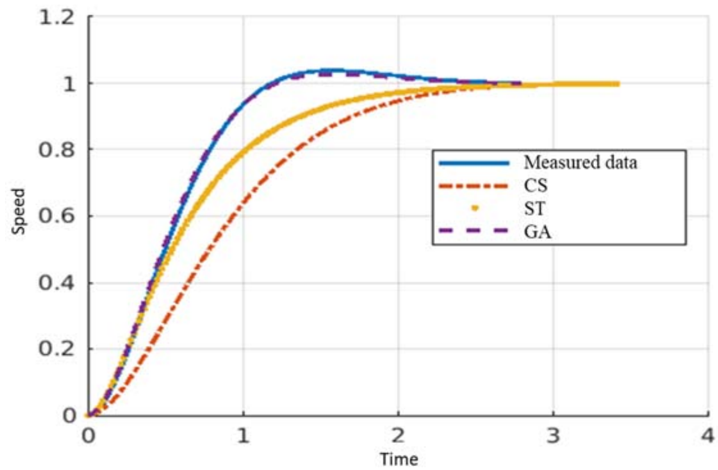


d) Genetic Algorithm (generation view)

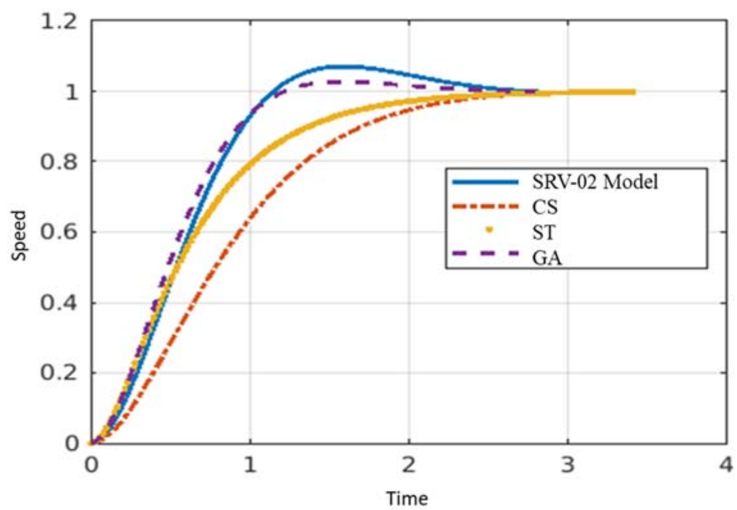
Figure 4: Behavior of each algorithm during the experiments carried out to find an approximate model of the system

Journal of Theoretical and Applied Information Technology
15th September 2020. Vol.98. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



a) *Measured data against optimization algorithms*



b) *Data from the SRV-02 model against the optimization algorithms*

Figure 5: Behavior of each algorithm during the experiments carried out to find an approximate model of the system

5. CONCLUSIONS

The graph in Figure 3b presents the experimental and simulated data of the QUANSER SRV-02 plant whose behavior is represented by a transfer function. However, between the model given by the manufacturer and the measured data there is an average error close to 5%, therefore, the measured data were used for the construction of the model presented. These data were used to search for a function that reproduces the behavior of the system when applying a unitary step function. In this case, equation 2 was used as a basis and its parameters were estimated using the techniques of exhaustive search optimization, simulated tempering and a genetic algorithm. It is worth mentioning that with all three techniques polynomials were found that converge to the expected value (see Figure 5).

The graphs in Figure 5 present the behavior of the best individuals found when running each optimization algorithm 10 times and in each execution 250 aptitude value evaluations were performed. Although, when executing the three techniques were found individuals that converge to a value, it is observed that the most effective technique was the genetic algorithm, since, when implementing it found at least one individual that has a convergence time similar to that expected with the transfer function and the experimental data. In addition, this individual also has the lowest average error (less than 1%) which is lower compared to the other two techniques.

As shown in the graphs in Figure 4, the technique that converged to a result more quickly was the genetic algorithm, since, when performing at least 120 assessments of the value of aptitude found sets of individuals that reduced the margin of error up to 3%. It should be noted that, when running this algorithm in subsequent generations are bad individuals, this is because a mutation was made that produced a large change in the population. However, the adaptability of the genetic algorithm eliminates the individual with the worst aptitude value thanks to the selection process. It should be noted that, when trying to reproduce an unknown data set by a genetic algorithm is possible to build a model that fits this data set with a margin of error less than 5%.

ACKNOWLEDGMENTS

This work was supported by the Universidad Distrital Francisco José de Caldas, in part through CIDC, and partly by the Technological Faculty. The views expressed in this paper are not necessarily endorsed by University. The authors thank the research group ARMOS for the evaluation carried out on prototypes of ideas and strategies.

REFERENCES

[1] S. Lee, K. Ryu, S. Nam, Y. Choi and H. Choi, "i-Manufacturing: Korean-style Manufacturing Innovation Strategy Promoting Balanced Regional Growth Based on Collaboration," *2008 International Conference on Smart Manufacturing Application*, Gyeonggi-do, 2008, pp. 58-62.

[2] N. Jiang, H. Lin, Z. Yin and C. Xi, "Research of paired industrial firewalls in defense-in-depth architecture of integrated manufacturing or production system," *2017 IEEE International Conference on Information and Automation (ICIA)*, Macau, 2017, pp. 523-526.

[3] B. Wang and Y. Zhu, "An analysis of the function of industrial engineering in equipment manufacturing industry based on DEA," *2011 IEEE 18th International Conference on Industrial Engineering and Engineering Management*, Changchun, 2011, pp. 668-671.

[4] Z. Ye, N. Cai and J. Zheng, "Organizational innovation of traditional manufacturing industry based on industrial evolution," *2009 IEEE International Conference on Industrial Engineering and Engineering Management*, Hong Kong, 2009, pp. 135-138.

[5] H. Watanabe, "Development of Wafer Transfer Simulator Based on Cellular Automata," in *IEEE Transactions on Semiconductor Manufacturing*, vol. 28, no. 3, pp. 283-288, Aug. 2015.

[6] U. Muhammad, B. R. Ferrer, W. M. Mohammed and J. L. M. Lastra, "An approach for implementing key performance indicators of a discrete manufacturing simulator based on the ISO 22400 standard," *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, St. Petersburg, 2018, pp. 629-636.

[7] D. Hacking, "Simulator design in advance of plant build," *IEE Colloquium on Operator Training Simulators*, London, UK, 1992, pp. 4/1-4/4.

[8] E. A. Zryumov, A. G. Zryumova and P. A. Zryumov, "Simulator of Carbon Nanotubes Manufacturing," *2019 International Science and Technology Conference "EastConf"*, Vladivostok, Russia, 2019, pp. 1-3.

[9] I. A. S. Abu Amra and A. Y. A. Maghari, "Forecasting Groundwater Production and Rain Amounts Using ARIMA-Hybrid ARIMA: Case Study of Deir El-Balah City in GAZA," *2018 International Conference on Promising Electronic Technologies (ICPET)*, Deir El-Balah, 2018, pp. 135-140.

[10] Y. Liu, Q. Zhao, H. Zhang, R. Xu, Y. Li and L. Wei, "A New Method to Predict RNA Secondary Structure Based on RNA Folding Simulation," in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 5, pp. 990-995, 1 September 2016.

[11] A. Nurhakim, N. Ismail, H. M. Saputra and S. Uyun, "Modified Fourth-Order Runge-Kutta Method Based on Trapezoid Approach," *2018 4th International Conference on Wireless and Telematics (ICWT)*, Nusa Dua, 2018, pp. 1-5.

[12] J. Latal et al., "Regresion models utilization for RSSI prediction of professional FSO link with regards to atmosphere phenomena," *2016 International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom)*, Graz, 2016, pp. 1-6.

[13] K. Natalia, K. Mikhail and B. Georgii, "On Using Gray Codes to Improve the Efficiency of the Parallel Exhaustive Search Algorithm for the Knapsack Problem," *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Saint Petersburg and Moscow, Russia*, 2019, pp. 1821-1825.

[14] G. Wang, C. Xu and G. Liu, "The transient electromagnetic inversion based on the simplex-simulated annealing algorithm," *2018 37th Chinese Control Conference (CCC)*, Wuhan, 2018, pp. 4321-4324.

[15] Y. Tominaga, Y. Okamoto, S. Wakao and S. Sato, "Binary-Based Topology Optimization of Magnetostatic Shielding by a Hybrid Evolutionary Algorithm Combining Genetic Algorithm and Extended Compact Genetic Algorithm," in *IEEE Transactions on Magnetics*, vol. 49, no. 5, pp. 2093-2096, May 2013.

[16] M. Shnan, T. H. Rassem and N. S. Zulkifli, “Facial image retrieval on semantic features using adaptive mean genetic algorithm”. *Telkomnika*, vol.17, no. 2, pp. 882-896, 2019.

[17] A. Gozali and S. Fujimura. “Reinforced Island Model Genetic Algorithm to Solve University Course Timetabling”. *Telkomnika*, vol.16, no. 6, pp. 2747-2755, 2018.

[18] A. Ariyani, W. Mahmudy, and Y. Anggodo, “Hybrid Genetic Algorithms and Simulated Annealing for Multi-trip Vehicle Routing Problem with Time Windows”. *International Journal of Electrical and Computer Engineering*, vol. 8, no. 6, pp. 4713-4723, 2018.

[19] J. Vinod and J. Prasad. “Solving N-queen Problem Using Genetic Algorithm by Advance Mutation Operator”. *International Journal of Electrical and Computer Engineering*, vol. 8, no. 6, pp. 4519-4523, 2018.

[20] QUANSER, QUARC Scripting for SRV02 Position Control Simulation, Apr 2020.