International Journal on Emerging Technologies **12**(2): 119-127(2021)

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

Genetic Algorithm Optimization of sliding Mode Controller Parameters for Robot Manipulator

***K. Saidi*[1], *A. Boumédiène*[2] *and D. Boubekeur*[3]**
[1]Ph.D. Student, GEE Department,
LAT, Laboratoire d'Automatique de Tlemcen, Université de Tlemcen 13000, Tlemcen, Algeria
[1]Asstistant Professor (Dept. of Automatic), Electrical Engineering Faculty,
Djillali Liabes University, Sidi Bel Abbes, Algeria.
[2]Professor, GEE Department, LAT, Laboratoire d'Automatique de Tlemcen,
Université de Tlemcen 13000, Tlemcen, Algeria.
[3]Lecturer, Oran 1 University, Oran, Algeria,
MELT, Manufacturing Engineering Laboratory of Tlemcen, Tlemcen, Algeria.

(Corresponding author: K. Saidi)
(Received 05 March 2021, Revised 25 April 2021, Accepted 29 May 2021)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: A combination between the artificial intelligence and the nonlinear controllers is proposed through this paper, for the control of robot manipulators with model uncertainties. The controller combines a genetic algorithm technique optimization with the so-called nonlinear sliding mode controller, SMC, approach. The motivation for using the sliding mode control in robotic control problems mainly relies on its appreciable features, like design simplicity and robustness. But the chattering effect can be destructive, which is typical of the conventional SMC. As compared to other optimization methods, the Genetic Algorithm method (GA) is a Metaheuristic based optimization algorithm approach to solve hard optimization problems, by simulating biological evolution and the fittest element principle in natural environment. Here, the genetic algorithm (GA) perspective to find the optimum control structure for a time horizon is presented and afterwards is tested on a control problem.For this proposed approach and to achieve this controller we must solve two problems, the first one is the elimination of the chattering phenomena, and the second one is to find the best parameters values of this controller. Robot manipulators are MIMO systems with coupled nonlinear dynamics and parametric variations. The implementation of the various control laws requires the determination of the various control parameters. This is not obvious since there is no direct method to determine these parameters for nonlinear systems. Our contribution is to introduce a method of metaheuristic optimization namely Genetic Algorithm in order to find the optimal parameters of nonlinear controllers.Numerical simulations using the dynamic model of a two-link planar rigid robot manipulator show the effectiveness of the proposed optimal control strategy based on SMC approach and GA in regulation and trajectory tracking problems.

Keywords: Robot manipulator, Genetic Algorithm, Nonlinear, Sliding Mode Control, Population, Optimization.

Abbreviations: The Genetic Algorithm, GA; Sliding mode controller, SMC; multiple-input–multiple-output, MIMO.

I. INTRODUCTION

The genetic algorithm (GA) has arisen from a desire to model the biological processes of natural selection and population genetics, with the original aim of designing autonomous learning and decision-making systems. Since its introduction, and subsequent popularization [5], the GA has been frequently used as an alternative optimization tool to conventional methods.

The application of GAs to control can entirely be divided into two distinct areas: off-line design and on-line optimization. Off-line applications have proved to be the most used and popular. On-line applications tend to be quite rare for several reasons and dependent to the difficulties associated with using a GA in real-time and directly influencing the operation of the system. GAs are applied in the control design application and to system identification. In each case, either the parameters of the controller or the structure of the systems can be optimized, or both at the same time. Other applications include stability analysis, fault diagnosis, sensor-actuator placement, and other combinatorial problems.

In recent years, enormous research efforts have been developed to perform the controllers for robot manipulator. Thus, robot manipulators have been successfully applied in various fields [27], e.g., space exploration, the medical domain, industrial appliances, etc. In general, robot manipulators are complex and highly coupled nonlinear systems with structured and unstructured uncertainties [7, 16], so it is difficult to establish an appropriate mathematical model for the design of a model-based control system [27]. The current trend of control approaches focuses on associating conventional control techniques like adaptive control, sliding mode control, etc. with artificial intelligent schemes, mainly fuzzy theory [9], neural network [27], genetic algorithm and other techniques in order to improve the performances of classical controllers in different aspects as well as to solve the gaps of traditional control techniques.

The design of robust adaptive controllers suitable for control of multiple-input–multiple-output (MIMO) nonlinear systems is one of the most challenging tasks for many control engineers, especially when complete knowledge of the system is not available. A robot manipulator is an uncertain nonlinear and coupled dynamic MIMO system, which suffers from structured and unstructured uncertainties such as payload variation, friction, external disturbances… etc. In the last few decades, many works of the association between the theories of classical control and the artificial intelligent control using genetic algorithm has undergone a rapid development to design a feedback controller for complex systems.

In theory and some case of practice many optimization methods were proposed, tested, and frequently used as well with purpose to solve such optimization problems, so we have the classical techniques like the gradient search method, the Newton methods and in other hand metaheuristics techniques, as the hill-climbing search, tabu search, iterated local search, GRASP, genetic algorithms [22, 34], scatter search, guided local search, ACO, PSO and many more.

The genetic algorithms technique can be applied to domains in which high complexity of the system and/or insufficient knowledge is there. Genetic algorithms can find global optimal solutions among the search space with the reproduction operators like crossover and mutation [5]. Then quickly find a reasonable solution to a complex problem, the genetic algorithms are very interesting and very effective techniques [22].

In the aspect of robust control performance, sliding mode control (SMC) is an attractive approach because it provides system dynamics with an invariance property to uncertainties once the system dynamics are controlled in the sliding mode [19].

As a powerful nonlinear controller, the Sliding Mode Controller (SMC), has been analysed by many researchers especially in recent years. The main reason to select this controller in wide range area is for it acceptable control performance [19]. However, this commonly used controller in wide range, however the classical sliding mode controller has some disadvantages. The major problem is chattering phenomenon [7], which can cause high frequency oscillation of the controller output. Another problem, is when the input signals are very close to the zero this controller is very sensitive to the noise effects [19]. Chattering phenomenon can cause in the case of the robot manipulators control some problems such as heat for mechanical parts and saturation. To eliminate or reduce the chattering, various works have been reported by many researchers and classified in two important methods, boundary layer saturation method and estimated uncertainties method.

Several approaches in this regard have already been proposed for fine-tuning robotic control systems via Genetic Algorithms in [28-31]. However, none has applied to optimise nonlinear controllers used in robot manipulator control. Also, the problematic encountered in the definition of sliding mode controller parameters is better suited for optimisation using GA instead of other optimisation tools. This problematic deal with having a reduced position error to the minimum attainable, while reducing the overshoot that usually appears at start-up.

In this paper, a nonlinear sliding mode controller is proposed to control the robot manipulator position with good performance and a better trajectory tracking. We extend our work by implementing genetic algorithm (GA) to search the optimal parameters (gains) of the controller to get the wished performances.

This paper is organized as follows: The robot manipulator description and structural properties are presented in Section 2. In Section 3, detailed design of sliding mode controller is introduced. Section 4 presents the theoretical aspect of genetic algorithm. Section 5 the application of the genetic algorithm to search the optimal parameters (gains) of the controller. Section 6 present the simulation results. At last, Section 7 draws the conclusions.

II. ROBOT MANIPULATOR DESCRIPTION AND STRUCTURAL PROPERTIES

The dynamics of a n-link robot manipulator can be described by a second-order nonlinear differential equation [16]

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + G(q) = \Gamma \quad (1)$$

Where $q, \dot{q}, \ddot{q} \in R^n$ are the link position, velocity, and acceleration vectors, respectively; $\Gamma \in R^n$ is the vector of applied link torques; $M(q) \in R^{n\times n}$ is the symmetric positive definite inertia matrix; $C(q,\dot{q})\dot{q} \in R^n$ is the Coriolis and centrifugal torque vector; $F_v \in R^{n\times n}$ is the diagonal matrix of the viscous friction coefficients; and $G(q) \in R^n$ is the gravity vector.

In the following, the structural properties of each term in the robot dynamics equation (1) are given [9]. These properties will offer a great deal of insight which will be used to derive robot control schemes [9, 16].

$$\text{P1:}\ \varphi_1 I_n \leq M(q) \leq \varphi_2 I_n \quad (2)$$

For some strictly positive constants φ_1 and φ_2.

$$\text{P2:}\ \dot{M}(q) - 2C(q,\dot{q}) \quad (3)$$

Is a skew symmetric matrix.

$$\text{P3:}\ \|G(q)\| \leq g_{max} \quad (4)$$

$$\text{P4:}\ \|F_v\dot{q} + F_s(\dot{q})\| \leq v_1\|\dot{q}\| + v_2 \quad (5)$$

Where $F_s(\dot{q})$ is the dry friction vector, with $v_1, v_2 > 0$

Notice that $\| \quad \|$ denotes the Euclidean vector norm.

friction coefficient and T_L is the load torque [38, 39, 40].

III. SLIDING MODE CONTROLLER

Sliding mode control (SMC) is a technique derived from variable structure theory. This technique of control has the capacity to manage nonlinear and time-varying systems. In the nonlinear SMC design (Slotine and all), for r = 2 (the relative degree) the proposed sliding surface $s(t)$ is given by the following equation [7, 10]:

$$s(t) = \dot{e}(t) + \lambda e(t) \quad (6)$$

Where $s(t)$ is an n×1 vector, λ is a diagonal positive definite constant matrix that determines the slope of the sliding surface and $e(t) = q(t) - q_d(t)$ is the tracking position error, in which $q_d(t)$ is the desired position trajectory and $\dot{e}(t) = \dot{q}(t) - \dot{q}_d(t)$ is the tracking velocity error in which $\dot{q}_d(t)$ is the desired speed trajectory.

The purpose of the sliding mode control is to design of a control law, such that the state vector e(t) remains on the sliding surface s(t) = 0 for all t ≥0. Therefore, it is required that the sliding surface is attractive, which means $lim_{t\to\infty} e(t) = 0$; then the error will converge to zero asymptotically. This implies that the system dynamics will track the desired trajectory asymptotically [19].

The derivative of the sliding surface is given by the following equation:

$$\dot{s}(t) = \ddot{e}(t) + \lambda\dot{e}(t) \quad (7)$$

Substituting (1) into (7) we obtain:

$$\dot{s}(t) = \ddot{q}(t) - \ddot{q}_d + \lambda(\dot{q}(t) - \dot{q}_d) \quad (8)$$
$$= M^{-1}(u - C.\dot{q}(t) - G) - \ddot{q}_d + \lambda(\dot{q}(t) - \dot{q}_d)$$

To achieve the desired performance, the solution of $\dot{s}(t) = 0$ gives us the relation of the control effort to be applied, which is called the equivalent control, and designed as $u_{eq}(t)$:

$$u_{eq}(t) = C.\dot{q}(t) + G + M[\ddot{q}_d - \lambda(\dot{q}(t) - \dot{q}_d)] \quad (9)$$

However, unpredictable perturbations from parameter variations or external load disturbances can occur, the equivalent control effort cannot ensure a favorable control performance. Hence, another control effort is added to eliminate the effect of the unpredictable perturbations. Which is referred as the reaching control effort represented by $u_d(t)$, given as follow [10]:

$$u_d(t) = -Ksign(s(t)) \quad (10)$$

Where $\mathrm{K} = \mathrm{diag}\{\mathrm{k}_1, \mathrm{k}_2, \dots, \mathrm{k}_\mathrm{n}\}$ represents reaching control gain, and $sign(.)$ is a sign function. For a 2 DOF robot manipulator, n=2 and $\mathrm{K} = \mathrm{diag}\{\mathrm{k}_1, \mathrm{k}_2\}$.

In total, the SMC law for nonlinearly uncertain systems that guarantees the stability and convergence can be represented as:

$$\mathrm{u}(\mathrm{t}) = u_{eq}(t) + \mathrm{u_d}(\mathrm{t}) = C.\dot{q}(t) + G + M[\ddot{\mathrm{q}}_\mathrm{d} - \lambda\,(\dot{\mathrm{q}}(\mathrm{t}) - \dot{\mathrm{q}}_\mathrm{d})] - \mathrm{K}\,\mathrm{sign}(\mathrm{s}(\mathrm{t})) \quad (11)$$

However, in the conventional SMC system, the sign function of the reaching control law will induce to chattering phenomena in the control efforts. This phenomenon may excite unmodeled high frequency modes, which degrades the performance of the system and may even lead to instability. That is why many procedures have been designed to reduce or eliminate this chattering. One of them consists in a regulation scheme in some neighbourhood of the switching surface, which in the simplest case, merely consists of replacing the sign function by a continuous approximation with a high gain in the boundary layer: for instance, sigmoid functions, saturation functions or tanh functions. Another solution to cope with chattering is based on the theory of higher-order sliding modes [1], [19].

In our case to solve this problem we propose to replace the sign function in the discontinuous controller by another function based on the tanh function.

IV. THE GENETIC ALGORITHM METHOD

Following his publication, "Adaptation in Natural and Artificial Systems", Holland is considered to be the founder of the Genetic Algorithms (GAs) method [12]. The GAs are based on the mechanism of natural selection and genetic reproduction [32]. A potential solution is being searched by GAs through a population of chromosomes. As analogy to the survival of the fittest law, fitness of each chromosome is evaluated within a population by the fitness (objective) function. Chromosomes with the highest fitness values will have a higher probability to survive and generate offspring. This allows GAs to improve or optimize its solution. GAs can be applied to solve nonlinear, discontinuous, multi-objective optimization problems [11, 22].

The structure of a GA is composed of an iterative procedure with the following five main steps [2, 25]:

A principle of coding of the element of population. This stage associates have each point of the space of state a structure of data. Two types of coding can be distinguished; the binary coding very much used and real coding.

To generate an initial population of fixed size, made of a whole finished of the solution, said initial generation. The choice of the initial population is significant because it can make more or less fast convergence towards the global optimum.

To define a function of evaluation (fitness) allowing to evaluate a solution and to compare it with the others. The development of a good function of adaptation (fitness function) must respect severalcriteria, which refer to its complexity like with the satisfaction of the constraints of the problem [18].

To generate new solutions using the genetic operators, these operators allow to diversify the population during generations and to explore the state space:

Selection: allows statistically to identify the best individuals of a population and to eliminate the bad ones. There are several methods of selection, among which one finds; roulette wheel mechanism, selection by tournament, and random selection [14].

Crossover operator: The crossover operator involves the exchange of genetic material between chromosomes (parents), in order to create new chromosomes (offspring) [20]. Various forms of this operator have been developed. The simplest form, single point crossover, is illustrated in Fig. 1 and two points in Fig. 2. This operator selects two parents, chooses a random position in the genetic coding, and exchange genetic information to the right of this point, thus creating two new offspring [15, 35]. It is applied with a probability Pc.

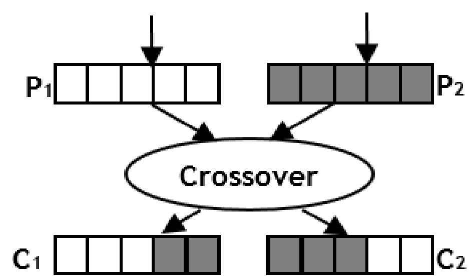

Fig.1. Single point crossover.

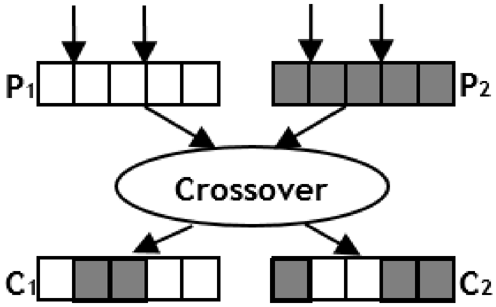

Fig. 2. Two point's crossover.

Mutation operator: Mutation is a process by which the chance of the genetic algorithm to reach the optimal point is reinforced through just an occasional alteration of a value at a randomly selected bit position. The process of mutation is simply to choose few members from the population pool according to the probability of mutation and to switch a 0 to 1 or 1 to 0 at a randomly selected mutation site on the selected string [19]. As shown in Fig. 3.

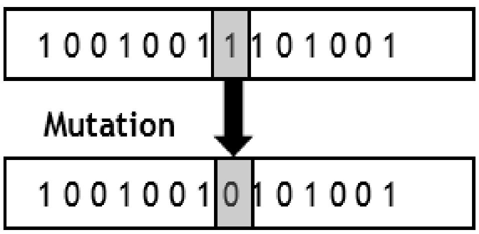

Fig. 3. Mutation operator.

Operator of elitism: With the creation of a new population, there are great chances that the best chromosomes are lost after the operations of crossover and mutation. To avoid that, one uses the method of elitism, which consists in copying one or more better chromosomes in the rising generation. Then, one generates the remainder of the population according to the usual algorithm of reproduction.

The test of stop plays a paramount role in the judgment of the quality of the individuals. Its goal is to ensure us optimality, of the final solution obtained by the genetic algorithm. The criteria of stops are of two natures:

Stop after a number fixed a priori of generations. It is the adopted solution when one duration maximum of computing time is imposed.

Stop when the population is not evolving or does not evolve sufficiently. We are then in the presence of a homogeneous population which one can think that it is located at the proximity of the optimum. This test of stop remains most objective and more used.

V. APPLYING THE GENETIC ALGORITHM

The conception of our controllers described by the equation (6) and (11) requires the specification of three parameters: λ which is the slope of the sliding surface, k1 and k2 the reaching control gains. These parameters must be chosen to ensure the convergence of the link positions to the desired positions. Unfortunately, wedon't have a direct method to find these parameters, because the nonlinearities and the coupling effects of the robotic systems. GA can be applied to obtain an optimal regulation, by considering the performance and the characteristics of the process, and taking into account all the responses of the system to be adjusted, the wished dynamic performance for the both links, minimal time of response and establishment, the static error zero, …etc [22]. In this section, the sliding mode controller will be optimized by a genetic algorithm.

The first step is to encode the problem into suitable GA chromosomes and after that constructs the population. All the parameters to be optimized of the controller are coded in binary with finished lengths. Character strings representing these parameters are concatenated and juxtaposed to build the chromosome like in Fig. 5. Every chromosome of the population can present a possible solution of the problem [3].

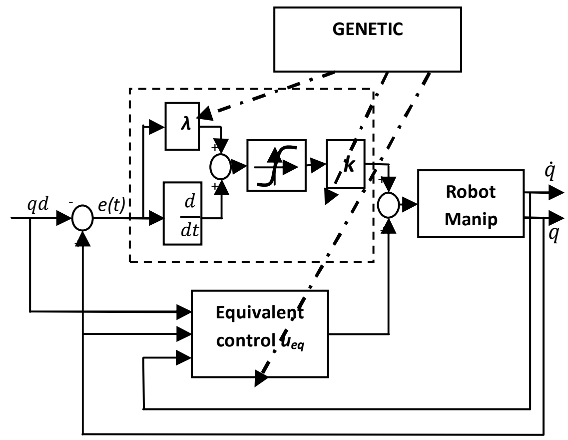

Fig. 4. The proposed optimized control scheme.

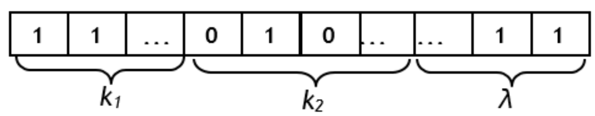

Fig. 5. The representation of the chromosome structure.

The choice of the initial population of individuals is a crucial step to reduce the execution time of the algorithm. Which is the size of the population which we can take? A too small population will evolve probably towards a little interesting local optimum. A too big population will be useless because the time of convergence will be excessive. The size of the population must be chosen to realize a good compromise meanwhile of calculation and quality of the result. Some works recommend 20 to 100 chromosomes in one population [15]. If we increase the number of chromosomes, we have more chance to have an optimal result. However, because we have to consider the execution time, we use between 80 and 100 chromosomes in each generation.

The fitness function can take many forms to evaluate the individuals of each generation; the most used are the sum of the error and the sum of the squared errors:

$$fitness = \frac{1}{T}\sum error \text{, or } fitness = \frac{1}{T}\sum error^2$$

We employ the maximum generation termination to stop the execution of the algorithm rather than considering the best chromosome fitness values changing rate because we want to manage the execution time. Therefore, we set 150 as the maximum generation. We choose Roulette Wheel Selection. After parents being selected, the crossover operation will be done. We use crossover two points because in our chromosome we have three parameters.

Mutation is done by setting mutation probability around 0.1%. In general, mutation operations should not be done too often because the searching process will change into random search as the mutation probability getting higher.

VI. SIMULATION RESULTS AND DISCUSSION

The controller of section 3 optimized by the algorithm genetic has been tested by simulations referring to the two degrees-of-freedom rigid robot manipulator described by the Figure 6 which is given by [8]:

$$M(q) = \begin{bmatrix} 8.77 + 1.02\cos(q_2) & 0.76 + 0.51\cos(q_2) \\ 0.76 + 0.51\cos(q_2) & 0.62 \end{bmatrix}$$

,

$$C(q,\dot{q}) = \begin{bmatrix} -0.5\sin(q_2)\dot{q}_2 & -0.5\sin(q_2)(\dot{q}_1+\dot{q}_2) \\ 0.5\sin(q_2)\dot{q}_1 & 0 \end{bmatrix}$$ and

$$G(q) = \begin{bmatrix} 7.6\sin(q_1) + 0.63\sin(q_1+q_2) \\ 0.63\sin(q_1+q_2) \end{bmatrix}$$

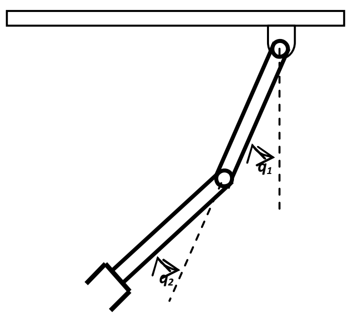

Fig. 6. Two degrees of freedom robot system.

A. Genetic algorithms applied to the sliding mode controller with initial condition [0.2, 0.1]

A robotic motion task is specified by defining a path along which the robot must move. Several interpolation functions can provide a trajectory such that $q(0) = q^{init}$ and $q(T) = q^{fin}$, where q^{init} and q^{fin} are respectively, the initial and the final configuration. We choose to use the fifth-degree polynomial interpolation, to ensure a smooth trajectory, which is continuous in positions, velocities and accelerations. The polynomial interpolation function is given by Khalil and all [4].

To generate the trajectory the desired values $q_1^{fin} = 1.1\,rd$, $q_2^{fin} = 1.3\,rd$ and $q_1^{init} = q_2^{init} = 0\,rd$ are used.

Table 1: GA Parameters for SMC Controller Optimization.

Population size	100
Number of generations	120
Crossover Fraction	0.8
Number of variable	3
Selection function	Roulette wheel mechanism

These parameters are determined through a series of experiments.

The values of the gain obtained are shown in Table 2:

Table 2: The best values of gains.

k_1	k_2	λ
750	203	13.5

Fig. 7 and Fig. 8 show the responses obtained for the two links with the same controller for the tracking objective (Section 4) with disturbance in input torque d=20% applied at t=1.2s. Both Fig. 9 and 10 represent the velocity of the two links, and Fig 11 and 12 the error evolution and the phase plane both position links respectively.

With an initial condition [0.2, 0.1].

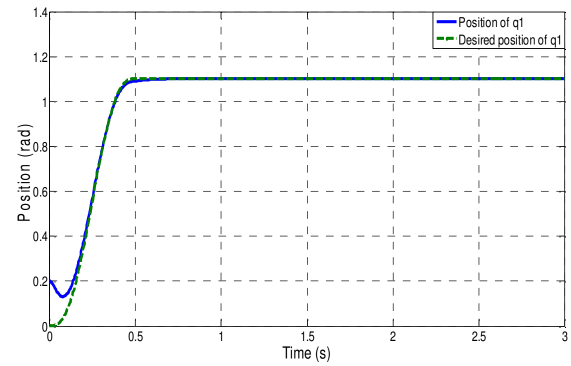

Fig. 7. Responses in tracking control for the first link.

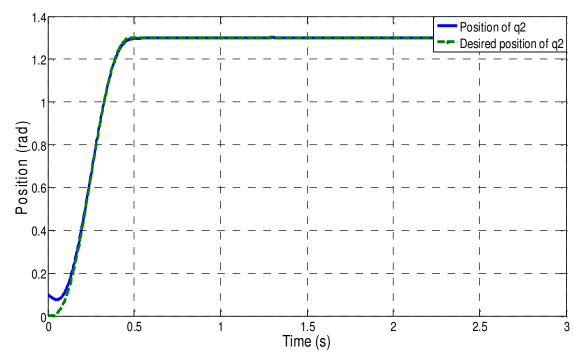

Fig. 8. Responses in tracking control for the second link.

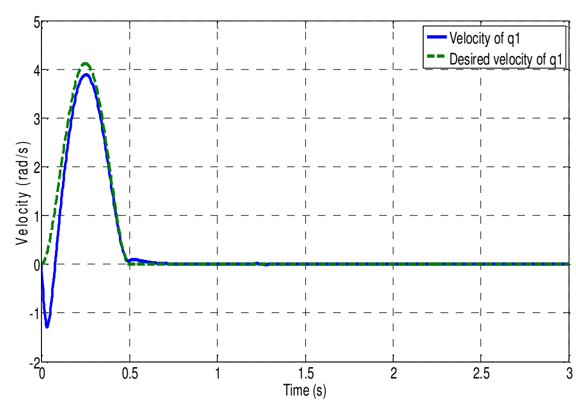

Fig. 9. Velocity of the first link.

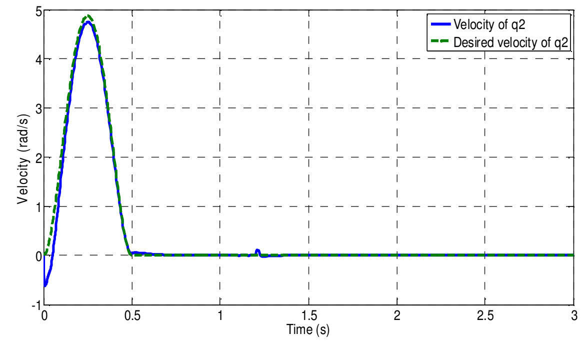

Fig. 10. Velocity of the second link.

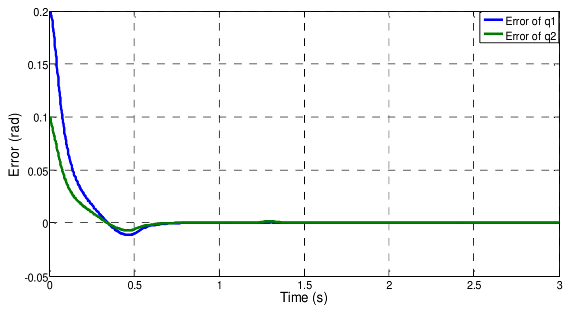

Fig. 11. Error evolution of both positions.

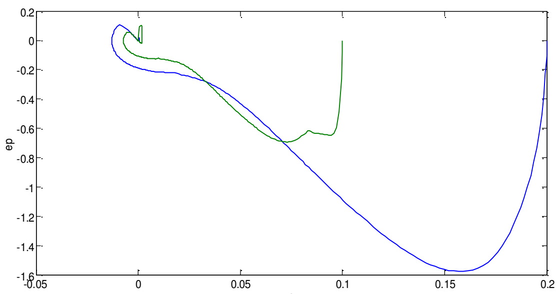

Fig. 12. Phase plane of both positions.

B. With initial condition [-0.1,-0.2]

For this step the same parameters are used, we change the initial condition to show the effectiveness of our controller. The initial conditions applied in this part are [-0.1, -0.2]

Fig. 13 and 14 show the responses obtained for the two links with the newest initials conditions. The Fig.15 and 16 represents the velocity of the two links, and the Fig. 17 and 18 represents the error evolution and the phase plane both position links respectively.

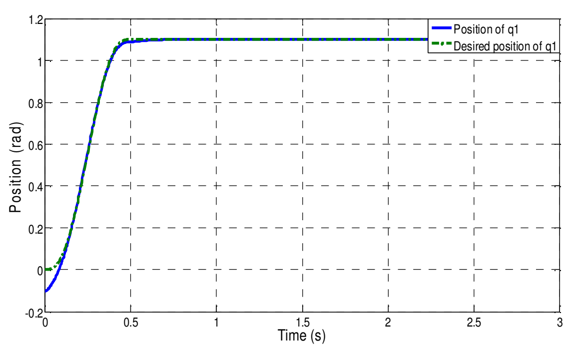

Fig. 13. Responses in tracking control for the first link with initial condition [-0.1,-0.2].

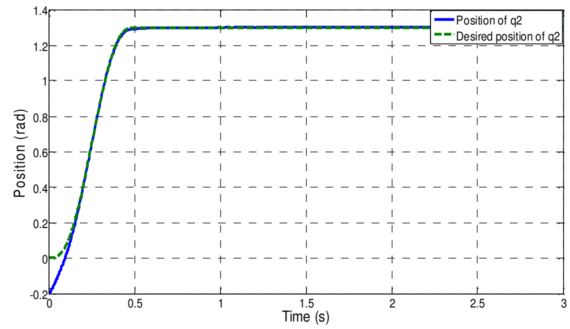

Fig. 14. Responses in tracking control for the second link with initial condition [-0.1,-0.2].

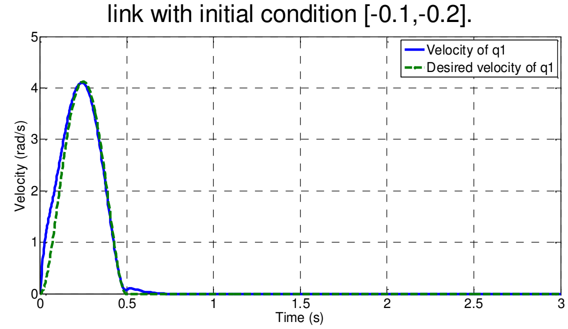

Fig. 15. Velocity of the first link.

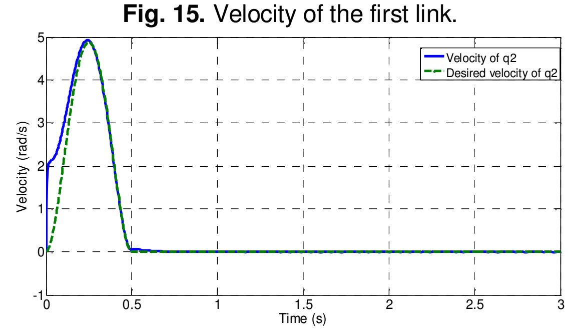

Fig. 16. Velocity of the second link

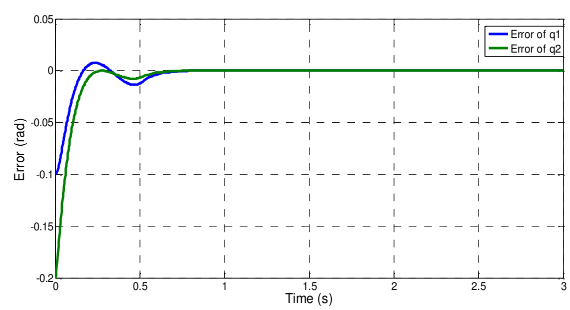

Fig. 17. Error evolution of both positions.

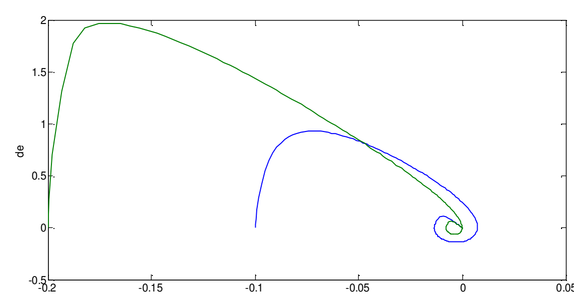

Fig. 18. Phase plane of both positions.

It can be seen that the proposed algorithm provides better closed-loop performance, through these simulations, in the tracking and regulation control for the robot manipulator, and allows us a facility to find the best parameters.
In Fig. 7-11 the results are obtained by using the optimal control parameters of each link angle after 120 iterations of the algorithm, where the population size used is 100. As shown in these figures, for the obtained parameters the responses of the both links have tracked the desired trajectories without overshoot, despite that the initial conditions of the both links are different of the initial points of the tracked trajectories, and as we can see the both errors converge to zero with a small time. It can be seen also that the disturbance applied at t =1.2s was rejected by the proposed controller.
To improve our algorithm, the Fig. 13-17 show the results by using the same optimal control, but the initial conditions are changed to a negative value for the both links, which mean that we can have a set points in the other side of the origin. The results show that we have a good trajectory tracking.
The choice of values 0.8 for the crossover operator and 0.01 for the mutation operator is justified to give a great diversity in the population resulting from these reproduction operations. To reproduce the chromosomes simple crossover and binary mutation are applied, with roulette wheel selection function.

From the results we can say that the GA could not find accurate solution using small population less than 20 chromosomes. It needs at least 20 to 30 chromosomes in population for achieving a better solution. It could also note that using very large population size (200 or 300 chromosomes) did not result in an improvement of the objective function values. In other side, when the population size increases it leads to increase of the needed computational resources like memory and time which can be a problem for large-scale tests. Therefore, we can affirm that populations with 80 to 100 individuals isoptimal with respect to the value of the fitness function and the needed computational resources.

In the Fig. 12 and 18, we present the phase plane analysis of the robot manipulator system. Besides allowing us to visually observe the motion patterns of our system, this will also help in the development and analysis of the nonlinear system, because a nonlinear system behaves similar to a linear system around each sliding surface and equilibrium point. As we can see from the Fig. 12 and 18 starting from the initial condition, the state trajectory reaches the time varying surface in a finite time, and then slides along the surface towards the desired positions.

In order to evaluate the performance of a closed-loop control system, a cost criterion can be set. The most common ones are in classical controller design methods like PID, the most common performance criteria are J_{IAE} (Integral of Absolute Error), J_{ISE} (Integral of Square Error), J_{ITAE} (Integral of Time-Weighted Absolute Error), and J_{ITSE} (Integral of Time-Weighted Square Error), They are given, respectively in (12) to (15)

$$J_{IAE} = \int_0^{tf} |e(t)|\, dt \tag{12}$$

$$J_{ISE} = \int_0^{tf} (e(t))^2\, dt \tag{13}$$

$$J_{ITAE} = \int_0^{tf} t|e(t)|\, dt \tag{14}$$

$$J_{ITSE} = \int_0^{tf} t(e(t))^2\, dt \tag{15}$$

Where tf is the simulation time. These four integral performance criteria in the frequency domain have their own advantage and disadvantages. For example, disadvantage of the J_{IAE} and J_{ISE} is that its minimization can result in a response with relatively small overshoot but a long settling time because the J_{ISE} performance criterion weights all errors equally independent of time. Furthermore, by using the J_{ITSE} performance criterion this tends to overcome the disadvantage of the J_{ISE} criterion.
We have tested in our work the four criteria cited above, since each criterion has advantages and disadvantages, in order to see the best criterion appropriate for our application, knowing that we are working on a MIMO system. The ISE criterion gave us satisfaction in terms of simulation time and quality of the results. A comparative study between the results obtained using all these criteria may be the subject of another paper.
The optimized SMC using GA perform better control specification such as, fast response and trajectory tracking task; also it can be seen that the optimized SMC with GA has not (or less) deviation compared with the others algorithms. The difficulty of the proposed approach compared to other works like [11] [28] [29] [30] is that one optimizes a nonlinear controller with several gains to be found, using several fitness functions.

VII. CONCLUSIONS

Refer to this paper, genetic optimization algorithm for nonlinear sliding mode controller (SMC) of two-link robot manipulators proposed.
The chattering phenomenon can be reduced by using linear saturation boundary layer function or tanh function in control law. The simulation results exhibit that the sliding mode controller with tanh function gives satisfying results.

Also the genetic algorithms (GA) are used to obtain the optimal sliding mode control parameters where the objective function for GA is based on the error. Hence, the majority of the applications that use this technique of optimization are, by nature, off-line. It should be noted that the use of genetic algorithms has several difficulties, especially for the choice of the population size and the generation number to be used, which determines the speed of the execution and the quality of results. By taking the greater value of the population size, we noticed that the execution algorithm time becomes longer which requires a powerful calculator, in other hand when we take a smaller value than 20 we have obtained bad results which means that we fell not a global optimum. The same GA process, which worked well in some situations, does not necessarily work well for other situations. Besides, it was a very difficult task to find suitable values of controller parameters through a fully trial and error procedure. Consequently, a more advanced optimal controller based on the application of GA, and a suitable cost function was proposed in this work. For the two-link robot manipulator control based on genetic algorithm, the simulation results show the good properties of the development schemes, in regulation and trajectory tracking applications.

In future work, the experimental implementation will show the effectiveness of this optimization technique. Other evolutionary techniques may further be implemented for the control parameters optimization.

VIII. FUTURE SCOPE

Future research could be carried out using other optimization techniques such as PSO, ant colony, etc. As well as the optimization of other control techniques.So,in our future works, several optimization methods will be compared todetermine the optimal one.
We can extend the application of these optimization methods to other controllers, such as backstepping and fuzzy logic controller.

ACKNOWLEDGEMENTS

This Works is supported by the Direction Generale de la Recherche Scientifique et du Developpement Technologique (DG RSDT).

Conflict of Interest: No conflict of interest

REFERENCES

[1]. Ahmed, F., Amer, E., Sallam, A. and Alawady, W. M., (2012). Quasi-sliding mode-based single input fuzzy self-tuning decoupled fuzzy PI control for robot manipulators with uncertainty. *Journal of robust and nonlinear control*, Vol. *22*: 2026-2054.

[2]. Poursamad, A. and Montazeri, M., (2008). Design of genetic-fuzzy control strategy for parallel hybrid electric vehicles. *Control Engineering Practice*, Vol. *16*(7): 861-873.

[3]. Amini, Y., Gerdroodbary, M. B., Pishvaie, M. R., Moradi, R. and Monfared, S. M., (2016). Optimal control of batch cooling crystallizers by using genetic algorithm. *Case Studies in Thermal Engineering*, *8*: 300-310.

[4]. Dombre, E. and Khalil, W., (2007). Modelling, Performance Analysis and Control of Robot Manipulators. First edition, revised by ISTE Ltd, London: British Library.

[5]. Goldberg, D.E., (1989). Genetic Algorithms in Search Optimizations and Machine Learning, 1st ed., Boston, MA, USA: Addison-Wesley Longman Publishing.

[6]. Ho, H. F., Wong, Y.K. and Rad, A.B., (2007). Robust fuzzy tracking control for robotic manipulators. *Simulation Modelling Practice and Theory*, Vol. *15*(7): 801-816.

[7]. Slotine, J.J.E. and Li, W., (1991). Applied Nonlinear Control. Prentice-Hall, *New Jersey: Englewood Cliffs*.

[8]. Khelfl, M. F., Zasadzinski, M., Darouach, M., Rafaralahy, H. and Richard, E., (1996). Reduced-order observer-based point-to-point and trajectory controllers for robot manipulators. *Control Eng Practice*, *4*(7): 991-1000.

[9]. Weil, Y., Jingdong, Z., Hou, L., Jia, F. and Chang, Q., (2013). Backstepping Adaptive Fuzzy Control for two-link robot manipulator", *International Journal of Computer Science Issues*, Vol.*10*(1).

[10]. Loucif, M., Boumediene, A., and Mechernene, A., (2015). Nonlinear Sliding Mode Power Control of DFIG under Wind Speed Variation and Grid Connexion. *Electrotehnica, Electronica, Automatica (EEA)*, *63*(3): 24-32.

[11]. Arunkumar, G., Gnanambal, I., Naresh, S., Karthik, P. C., and Patra, J. K., (2015). Parameter Optimization of Three-Phase Boost Inverter Using Genetic Algorithm for Linear Loads. *Proceedings of the 5th International Conference on Advances in Energy Research (ICAER), Mumbai, India, 15-17, Energy Procedia*, *90*: 559-565.

[12]. Roberge, V., Tarbouchi, M. and Okou, F., (2014). Strategies to Accelerate Harmonic Minimization in Multilevel Inverters Using a Parallel Genetic Algorithm on Graphical Processing Unit. IEEE Transactions on Power Electronics, *29*(10): 5087-5090.

[13]. Miguel, A., Llama, Santibaliezt, V., Kelly, R., and Florest, J., (1999). Stable Fuzzy Self tuning Computed-torque Control of Robot Manipulators. Proceedings of the 1998 IEEE International Conference on Robotics and Automation Leuven, Belgium.

[14]. Ar, Y., Bostanci, E., (2016). A genetic algorithm solution to the collaborative filtering problem. *Expert Systems with Applications*, *61*: 122-128.

[15]. Fleming, P. J., and Purshouse, R. C., (2001). Genetic Algorithms In Control Systems Engineering. Department of Automatic Control and Systems Engineering, University of Sheffield, S1 3JD, UK, Research Report No. 789.

[16]. Spong, M.W., and Vidyasagar, M., (1989). Robot dynamics and control. *New York: John Willey and Sons*.

[17]. Vidyasagar, M., (1993). Nonlinear systems analysis. Second Edition, USA: *Prentice-Hall International*.

[18]. Utkin, V. I., (1977). Variable structure systems with sliding modes. *IEEE Trans. on Aut. Cont.*, *22*: 211-222.

[19]. PERRUQUETTI, W., and BARBOT, J. P., (2002). Sliding Mode Control in Engineering. USA: Marcel Dekker.

[20]. Irshad, M., Khalid, S., Hussain, M. Z., and Sarfraz, M., (2016). Outline capturing using rational functions with the help of genetic algorithm. *Applied Mathematics and Computation*, *274*: 661-678.

[21]. Ravi, Y., and Balakrishnan, P. A., (2011). Stable Self Tuning Genetic Fuzzy Temperature Controller for Plastic Extrusion System. *International Journal of Reviews in Computing*, *5*: 21-28.

[22]. Saidi, K., and Allad, M., (2015). Fuzzy controller parameters optimization by using genetic algorithm for the control of inverted pendulum. *Proceedings of the 3rd International Conference on Control, Engineering and Information Technology (CEIT), Tlemcen, Algeria.*

[23]. SAHRAOUI, K., Kouzi, K., and Ameur, A., (2017). Optimization of MRAS based Speed Estimation for Speed Sensorless Control of DSIM via Genetic Algorithm. *Electrotehnica, Electronica, Automatica (EEA)*, *65*(3): 156-162.

[24]. Houari, H., Hassem, A., Souier, M., and Sari, Z., (2017). Development of Genetic Algorithm for the Improvement of the Performances of Flexible Manufacturing Systems (FMS). *Electrotehnica, Electronica, Automatica (EEA), 65*(3): 197-204.

[25]. Reboucas Filho, P.P., Da Silva, S. P., Praxesdes, V.N., Hemanth, J., and Dealbuquerque, V. H. C., (2019). Control of singularity trajectory tracking for robotic manipulator by genetic algorithms. *Journal of Computational Science*, *30*: 55-64.

[26]. Boubekeur, D., Boumediene, A., Sari, Z.,and TAHRAOUI, S., (2015). A Control Comparative Study of

an Electric Powered Wheelchair System. *Electrotehnica, Electronica, Automatica (EEA), 63*(2): 86-94.
[27]. Wai, R. J., Muthusamy, R., (2013). Fuzzy-Neural-Network Inherited Sliding-Mode Control for Robot Manipulator Including Actuator Dynamics. *IEEE Transactions on Neural Networks and Learning Systems*, *24*(2): 274-287.
[28]. Refoufi, S. and Benmahammed, K., (2018). Control of a manipulator robot by neuro-fuzzy subsets form approach control optimized by the genetic algorithms. *ISA Transactions, 77*: 133-145.
[29]. Pizarro-Lerma, A.O., García-Hernández, R. and Santibáñez, V.A., (2018). Fine-Tuning of a Fuzzy Computed-Torque Control for a 2-DOF Robot via Genetic Algorithms. IFAC-Papers Online, *51*(13): 326-331.
[30]. Bakdi, A., Hentout, A., Boutami, H., Maoudj, A., Hachour, O. and Bouzouia, B., (2017). Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. Robotics and Autonomous Systems, *89*: 95–109.
[31]. Sangdani, M.H., Tavakolpour-Saleh, A.R. and Lotfavar, A., (2018). Genetic algorithm-based optimal computed torque control of a vision-based tracker robot: Simulation and experiment. *Engineering Applications of Artificial Intelligence*, *67*: 24–38.
[32]. Kulkarni, J., and Bichkar, R. (2020). Image Fusion using Hybrid Genetic Algorithm. *International Journal on Emerging Technologies*, *11*(3): 442-447.
[33]. Tensing, D., Vincent Sam Jebadurai S., Paulson, J., Rahul A. V. and Jefrey, K. (2020). Optimization in the Design of Prestressed Concrete Girders using Excel and MATLAB. *International Journal on Emerging Technologies, 11*(2): 420–426.
[34]. Harish Kumar, P., and Mageshvaran, R. (2020). Optimization Techniques and Algorithms for DG placement in Distribution System: A Review. *International Journal on Emerging Technologies*, *11*(3): 141–157.
[35]. Tiwari, R., and Kumar Yawale, S. (2016). Review of Several Optimization Techniques for Control of Induction Motor. *International Journal of Electrical, Electronics and Computer Engineering, 5*(2): 37-40.

How to cite this article: Saidi, K., Boumédiène, A. and Boubekeur, D. (2021). Genetic Algorithm Optimization of sliding Mode Controller Parameters for Robot Manipulator. *International Journal on Emerging Technologies, 12*(2): 119–127.