International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 56, No. 1, February, 2023, pp. 89-96
Journal homepage: http://iieta.org/journals/jesa

An Optimal GA-Based Backstepping Control Scheme for a MIMO Nonlinear System

Khayreddine Saidi[1,2*], Abdelmadjid Boumediene[2], Djamila Boubekeur[3,4]

[1] Electrical Engineering faculty, Djillali Liabes University, Sidi Bel Abbes 22000, Algeria
[2] LAT, Laboratoire d'Automatique de Tlemcen, Université de Tlemcen, Tlemcen 13000, Algeria
[3] MELT, Manufacturing Engineering Laboratory of Tlemcen, Tlemcen University, Tlemcen 13000, Algeria
[4] Institut des Sciences et Techniques Appliquées, Université Oran1, Oran 31000, Algeria

Corresponding Author Email: saidi_kheiro@yahoo.fr

https://doi.org/10.18280/jesa.560112

Received: 3 February 2023
Accepted: 13 February 2023

Keywords:
nonlinear system, robot manipulator, genetic algorithm, backstepping control, optimization, population

ABSTRACT

In the presence of external disturbances, a backstepping control scheme based a genetic algorithm (GA) is built with the objective of tracking a desired trajectory of robot manipulators. The nonlinear-coupled higher order dynamic model of an n-link robot manipulator is first briefly presented. Then, a traditional backstepping control system (BSC) creates a two-link robotic manipulator position tracking control. Furthermore, a mix of nonlinear control and artificial intelligence is suggested for manipulator robot control. To determine the optimal control parameters, the backstepping controller is combined with the genetic algorithm, a metaheuristics-based optimization technique. The effectiveness of the suggested optimal control strategy based on backstepping approach and GA in trajectory tracking problems, such as a good angle tracking and a good disturbance rejection capability, is demonstrated by numerical simulations using the dynamic model of a two-link planar rigid robot manipulator.

1. INTRODUCTION

A number of studies have been conducted in recent years to perform the controllers for robot arms, which are successfully used in many domains [1], including medical domain, automotive industry, space exploration, and many other industrial applications. Control problems of robotic manipulators can be formulated as the determination of the generalized forces evolution of which the actuators must exert, while satisfying certain performance criteria to ensure the execution of a task whatever operating conditions. For example, in robotic systems used for assembly operations with heavy parts carried by the manipulator clamp, a fast-functioning robot, etc.

Robot manipulators typically have unstructured and structured uncertainties including friction, payload variation, external disturbances, etc., and are highly coupled, complex, MIMO and nonlinear systems [2, 3]. Therefore, it is challenging to develop a suitable mathematical model, which is required for the construction of a controller [1].

To address this issue, the current trend in control approaches is to combine traditional control techniques, such as sliding mode control, adaptive control, backstepping control, etc., with artificial intelligence schemes, primarily neural network [1], fuzzy logic theory [4], genetic algorithm and other techniques. In order to enhance the performances of conventional controllers in various characteristics as well as to close the gaps left by traditional controllers.

Many studies on the combination between classical control theory and artificial intelligence approaches have advanced quickly in the recent few decades in order to construct feedback controllers for complicated systems.

Due to its straightforward design and reliable performance, backstepping control is one trustworthy control technique that has lately been deployed. It's a control strategy that entails stepping back from the process's control input. This technique is considered as an effective and interesting tool that can be used to build controllers for nonlinear systems with uncertain parameters and transform them into strict-feedback forms [5].

For this purpose, numerous studies have been carried out in [6, 7]. This control technique has specifically demonstrated its effectiveness in the control of robot manipulators in [8, 9].

However, a drawback of all this control approaches employed within the previous cited contributions, known as "explosion of complexity" as the system size is increased [10].

Designing a Backstepping control for the robotic manipulator in our situation necessitates precise knowledge of the system model, which is challenging when complexity increases due to increasing degrees of freedom (DoF) in advanced manipulators case, and their high-coupled nonlinear models with structured and unstructured uncertainties.

The genetic algorithm (GA) is among solutions of this problem kind. Interest in GA has grown with the goal to design systems with autonomous learning and decision-making, by modelling genetic properties of species and the natural selection phenomenon in the biological processes. Since its appearance and ensuing expansion [11], the GA has been considered as an interesting optimization technique that can replace traditional approaches with success.

Many optimization techniques were proposed, evaluated, and regularly employed in theory and some cases of practice with the aim of solving such optimization problems. The Newton methods and gradient search methods are examples of classical techniques. On the other hand, there are metaheuristic techniques like tabu search, hill-climbing search, ROA, WOA, GA [12], simulated annealing, scatter search, ACO, PSO [13] and much more.

Two distinct areas of the GAs applications in control field can be mentioned: the off-line optimization and the online design. Applications with offline optimization have proven to be the most widely utilized and adopted. For a variety of reasons, including the difficulties involved in employing a GA in real-time and how directly they affect the way a process works, there are not many online applications for GAs. GAs are used in the system identification applications and specially to control design. In each situation, either the controller parameters or the systems structure can be optimized, or both at once. Stability analysis, sensor-actuator placement, fault diagnostics, and a number of other issues are only a few examples of the many applications.

Therefore, the reproduction operators such as mutation and crossover allow the GAs to identify global optimal solutions inside the search space [11]. Then rapidly identify a reasonable solution to a problem with high complexity, the GAs are highly interesting and powerful tools [12].

The implementation of the different control laws requires the determination of the different parameters of these controllers. This is not obvious since there is no direct method to determine these parameters. Our contribution in this work is to introduce a metaheuristic optimization method known as Genetic Algorithm in order to find the optimal parameters of the nonlinear controller.

After the introduction section, this paper is organized as follows: The second section covers a mathematical description of our system. After that, a Backstepping controller is proposed to control the robot manipulator position in section 3. Our work is prolonged by GA implementation to find the best controller parameters (optimal gains) to get the wished performances in section 4 and 5. Simulation tests and discussions are presented in section 6. The conclusions are finally drawn in the last section, Section 7.

2. MATHEMATICAL DESCRIPTION OF THE MIMO SYSTEM

An n-dof robot manipulator's dynamics in its joint space can be described by a second-order nonlinear differential equation [3]

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

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

Following are the structural properties for each term in the robot dynamics mentioned in Eq. (1) [4]. These properties will provide a wealth of knowledge that will be helpful in developing robot control strategies [3, 4].

$$\text{P1:}\varphi_1 I_n \le M(q) \le \varphi_2 I_n \qquad (2)$$

For some strictly positive constants $\varphi_1$and φ_2.

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

Is a matrix with skew symmetry.

$$\text{P3:}\|G(q)\| \le g_{max} \qquad (4)$$

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

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

Consider the fact that $\|\quad\|$ represents the Euclidean vector norm.

The objective sought through this paper is to develop a controller to have a good trajectory tracking, between the output vector and the desired trajectory q_d. With the aim to simplify the control design formulation, we define $x_1 = q, x_2 = \ \dot{q}$, the formulation (1) can be denoted by (6)

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = M^{-1}(x_1) * (\Gamma - C(x_1,x_2)x_2 - F_v x_2 - G(x_1)) \\ y = x_1 \end{cases} \qquad (6)$$

where, $y \in R^n$ is the system output.

3. BACKSTEPPING CONTROL LAW APPLIED TO TRAJECTORY TRACKING

Backstepping control technique is recognized as a recursive Lyapunov-based technique. The principal idea of this technique is to design a controller in recursive way by considering as 'virtual controls' some state variables, and uses them as intermediate control laws.

As previously mentioned, in tracking systems, the sought control objective is to create a law of control that forces the joint position q to follow the desired position q_d. The Backstepping control scheme that is used to achieve the position-tracking objective may be described down into many steps as shown below.

1st step:

In the position tracking case, the error of tracking is chosen as the first regulatory variable defined as:

$$\varepsilon_1 = x_1 - x_{1d} \qquad (7)$$

And its derivative is:

$$\dot{\varepsilon}_1 = \dot{x}_1 - \dot{x}_{1d} \qquad (8)$$

The choice of the first Lyapunov function is given by following:

$$V_1 = \frac{1}{2}\varepsilon_1^T\varepsilon_1 \qquad (9)$$

Then the derivative of V_1 given in Eq. (9) can be represented as:

$$\dot{V}_1 = \varepsilon_1^T\dot{\varepsilon}_1 = \varepsilon_1^T(\dot{x}_1 - \dot{x}_{1d}) \qquad (10)$$

Assuming that $\dot{x}_1$ is the controlling term in Eq. (8), a virtual control α_1 is selected to stabilize the subsystem so that $\dot{x}_1 = \alpha_1$. By using the Lyapunov function V_1, α_1 is derived as:

$$\alpha_1 = -K_1\varepsilon_1 + \dot{x}_{1d} \qquad (11)$$

where, $K_1 \in R^{n\times n}$ is a positive-definite diagonal matrix. Therefore, by substituting (11) in (10), the time derivative of the first Lyapunov function V_1 can be represented as:

$$\dot{V}_1 = -\varepsilon_1^T K_1 \varepsilon_1 \tag{12}$$

2nd step:

The difference between α_1 and $\dot{x}_1$ defines the next regulatory variable ε_2, given that, the actual control has not yet been applied and that these two terms will not be equal. Thus, ε_2 is provided by

$$\varepsilon_2 = \dot{x}_1 - \alpha_1 = x_2 + K_1\varepsilon_1 - \dot{x}_{1d} \tag{13}$$

The derivative of ε_2 given in Eq. (13) can be expressed as

$$\dot{\varepsilon}_2 = \dot{x}_2 - \ddot{x}_{1d} + K_1\dot{\varepsilon}_1 \tag{14}$$

The second Lyapunov function V_2 is chosen as

$$V_2 = V_1 + \frac{1}{2}\varepsilon_2^T\varepsilon_2 \tag{15}$$

Then the derivative of V_2 expressed by (15) can be represented as

$$\dot{V}_2 = \dot{V}_1 + \varepsilon_2^T\dot{\varepsilon}_2 \tag{16}$$

By replacing (6), (12) and (14) in (16), the derivative of the function of Lyapunov V_2 can be represented as

$$\begin{aligned}\dot{V}_2 &= -\varepsilon_1^T K_1\varepsilon_1 + \varepsilon_2^T(\dot{x}_2 - \dot{\alpha}_1) \\ &= -\varepsilon_1^T K_1\varepsilon_1 + \varepsilon_2^T(M^{-1}(x_1) * (\mathrm{U} - C(x_1,x_2)x_2 - \\ &\quad F_v x_2 - \quad G(x_1)) - \dot{\alpha}_1)\end{aligned} \tag{17}$$

The following structure is expected for the backstepping control law U_{BTP}:

$$\begin{aligned}U_{BTP} = C(x_1,x_2)x_2 + G(x_1) + M(x_1)\,\dot{\alpha}_1 \\ - K_2\,M(x_1)\,\varepsilon_2\end{aligned} \tag{18}$$

where, $K_2 \in R^{n\times n}$ is a positive-definite diagonal matrix. Consequently, using (16)-(18), the time derivative of V_2 can be written as

$$\begin{aligned}\dot{V}_2 = \dot{V}_1 - \varepsilon_2^T\,K_2\,\varepsilon_2 \\ \leq -\varepsilon_1^T K_1\varepsilon_1 - \varepsilon_2^T\,K_2\,\varepsilon_2 \;\leq 0\end{aligned} \tag{19}$$

As can be seen from (19), $\dot{V}_2$ is a negative-definite function. Backstepping control system stability is guaranteed by using the Lyapunov stability theorem [2, 14].

4. THE GA METHOD

Genetic Algorithms (GAs) are metaheuristic methods which can be used to solve different optimization problems [12, 15]. Holland through his work entitled, ‘‘Adaptation in Natural and Artificial Systems’’ introduce for the first time the principles of GAs method [16]. They are based on the genetic reproduction of biological processes, thus the GAs implement the optimization strategies by using the simulation of evolution of species through the process of naturel selection.

Based on using a direct analogy of natural and biological behaviour, GAs work with the principle of evaluating of each individual in a given population, every individual is represented by a chromosome, and evaluated by using a function of fitness. Chromosomes with the most important fitness function will have an important chance to participate in the next generation and survive to generate offspring.

An iterative procedure of five steps is composed to form a GA structure, these steps are described below [17, 18]:

Coding: In this step, the parameters as a set, which can be a potential solution to a given problem are associated together to form a chromosome (string of values). Generally, the best is to use a binary coding; however, there are other kinds of coding that can be identified, such as real coding.

Generate an initial population: The size of the initial population must be fixed. The choice of the size is very important because it conditions that the convergence towards the global optimum is fast or less.

The fitness function (function of evaluation): Defined to be proportional to the ability or utility of the chromosome, to evaluate and compare every solution to the other one. The fitness function must be developed with respect of several criteria, by taking into account the satisfying of the constraints existing in the problem [19, 20].

A whole new population of solutions is generated by using the genetic operators, which can allow a population diversifying and an exploration of the state space, below the genetic operator:

Selection: Through this operator, the best chromosomes (individuals) are selected from the current population. This operator gives a favour in the selection as parents, to the best individuals for producing offspring. Random selection, roulette wheel mechanism and selection by tournament are the most used methods of selection [21].

Crossover operator: This operator choose two individuals (parents “Par”) with a probability Pc, takes their chromosome strings, cuts them at a chosen random position and swap the tail segment to the right of this point, to produce two offspring (Child “Ch”) [22, 23]. Several forms of crossover operator exist, the single point crossover showed in Figure 1 and two points in Figure 2.

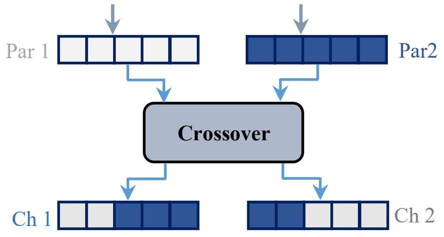

Figure 1. Single point crossover

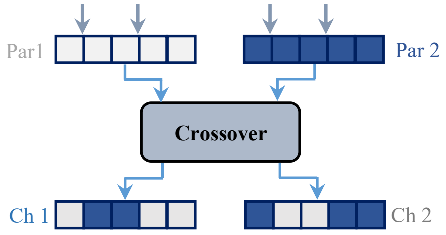

Figure 2. Two point’s crossover

Mutation operator: This operator is applied after the crossover operator to reinforce the GA to reach the optimal solution. It simply select randomly a bit position on the chosen chromosome and alters the value (1 to 0 or 0 to 1) with a probability (typically small 0.001) [24]. Figure 3 shows the eighth bit of chromosome being mutated.

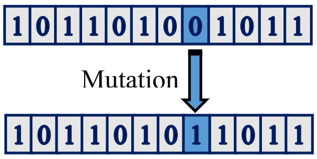

Figure 3. Mutation operator

Elitism Operator: After the application of the crossover and mutation operators, a great risk exists to lose the best chromosomes. One or more of the best chromosomes are copied as part of the elitism operator to participate in the next generation, and then according to the usual reproduction algorithm, the population remainder is generated.

The algorithm stop test plays an important part in the judgment of the individual's quality. This stop test allows us to ensure that the solution obtained by the GA is optimal; two types of the stop test exist:

The first one is to stop the algorithm after a fixed number of generations, defined a priori. This criteria of stop is adopted when a computing time duration is imposed.

The second one is to stop the algorithm when there is no evolving or the population does not evolve sufficiently. Then the population is homogeneous and is close to the optimal solution, this stop test is the most objective and the most used.

5. CONTROLLER OPTIMIZATION BY USING GA

It is necessary to specify two parameter matrices K_1 and K_2 in order to construct our Backstepping controller as specified by Eqns. (11) and (18). The choice of these parameters must guarantee the convergence of the link positions to the desired positions. Unfortunately, due to the nonlinearities and coupling effects of the robotic systems, we don't have a direct technique to determine these parameters. GA can be used to find the optimal parameter values as presented in Figure 4, by taking into account the process's performance and characteristics, as well as all the system's potential responses, the desired dynamic performance for both links, the shortest response and establishment times, the static error zero, etc. [12]. As can be seen in Figure 4 the genetic algorithm based on errors as inputs to build the objective function, acts to optimize the backstepping controller parameters.

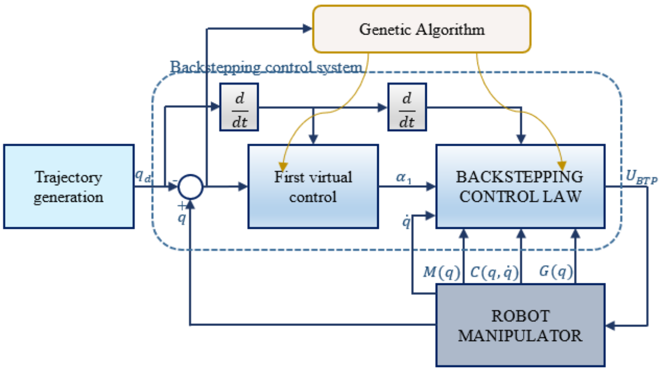

Figure 4. The proposed optimized control scheme

By using a binary code, all the controller parameters that need to be optimized are transformed with a finished length. In this first step, the problem is encoded into appropriate GA chromosomes, to build population after that. Strings and character that represent these parameters are juxtaposed and concatenated to construct the chromosome, as represented in Figure 5. Each chromosome belonging to the population may be a potential problem solution [25].

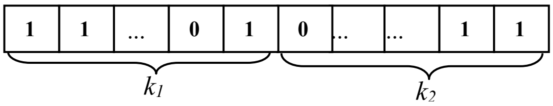

Figure 5. The representation of the chromosome structure

A critical step in reducing the algorithm's execution time is selecting the initial population. Which is the population's size, which can be taken? By taking a small population size, it will progress probably to a local optimum with a little interest. Otherwise, a big population will be unusable for the reason that the convergence time could be excessive. The population size choice must ensure a reasonable balance between calculation time and result quality. Many works suggest in one population between 20 and 100 chromosomes [22]. We have a better chance of getting the optimal result, if we have more chromosomes. Nevertheless, we must consider the execution time. We therefore use between 80 and 100 as the number of chromosomes for each generation.

A cost criterion can be set, with the objective to assess the closed-loop control system performances. These criteria are also used with conventional methods for controller design such as PID controller. The most frequently employed criteria of performance are JISE (Integral of Square Error), JIAE (Integral of Absolute Error), JITSE (Integral of Time-Weighted Square Error), and JITAE (Integral of Time-Weighted Absolute Error). Consequently, the fitness function can be expressed in a variety of forms to assess individuals at each generation; the most employed are as follows:

$$J_{IAE} = \int_0^{tf} |e(t)|\, dt \tag{20}$$

$$J_{ISE} = \int_0^{tf} (e(t))^2\, dt \tag{21}$$

$$J_{ITAE} = \int_0^{tf} t|e(t)|\, dt \tag{22}$$

$$J_{ITSE} = \int_0^{tf} t(e(t))^2\, dt \tag{23}$$

where, tf is the simulation final time. The use of one or other of these four integral performance criteria may have advantages and disadvantages. For example, as disadvantage, the minimization by using JIAE and JISE performance criterion can result a relatively small overshoot in the response but a lengthy settling time because the JISE weights all errors equally time independent. Besides, employing the JITSE criterion of performance, this leads to overcome the JISE criterion's disadvantage.

In order to control the time of the algorithm execution, we use the maximum generation termination as opposed to the optimal chromosome fitness values that change rate. Therefore, the maximum number of generation is set at 150, and we opt to the Roulette Wheel Selection. Once the parents selection step is made, a simple crossover point operation is applied.

The mutation operator is done by setting its probability around 0.1%. Generally, this operator shouldn't be used too frequently because the search process will become a random search as the mutation probability increases.

6. RESULTS OF SIMULATION AND DISCUSSION

Simulations using the 2-dof rigid robot manipulator seen in Figure 6 were used to test the controller of section 3 that was tuned by the genetic algorithm. The dynamics of this robot are expressed as in Eq. (1). The robot dynamics are stated as follows by taking into account the symbols in Table 1 [26]:

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (24)$$

The elements $M_{pk}(q)(p,k=1,2)$ from the inertia matrix $M(q)$ are expressed as:

$$\begin{gathered} M_{11} = m_1 l_{c1}^2 + m_2\left(l_1^2 + l_{c2}^2 + 2 l_1 l_{c2} cos(q_2)\right) + I_1 \\ + I_2 \\ M_{12} = M_{21} = m_2 l_1 l_{c2} cos(q_2) + m_2 l_{c2}^2 + I_2 \\ M_{22} = m_2 l_{c2}^2 + I_2 \end{gathered} \quad (25)$$

The elements $C_{pk}(q,\dot{q})(p,k=1,2)$ of the centrifugal and Coriolis matrix $C(q,\dot{q})$ are given as follows:

$$\begin{gathered} C_{11} = -m_2 l_1 l_{c2} sin(q_2)\dot{q}_2 \\ C_{12} = -m_2 l_1 l_{c2} sin(q_2)(\dot{q}_1 + \dot{q}_2) \\ C_{21} = -m_2 l_1 l_{c2} sin(q_2)\dot{q}_1 \\ C_{22} = 0 \end{gathered} \quad (26)$$

The gravitational torque vector $G(q)$ is given by the following expression:

$$\begin{gathered} g_1 = (m_1 l_{c1} + m_2 l_1) g \ cos(q_1) \\ + m_2 l_{c2} g \ cos(q_1 + q_2) \\ g_2 = m_2 l_{c2} \ cos(q_1 + q_2) \end{gathered} \quad (27)$$

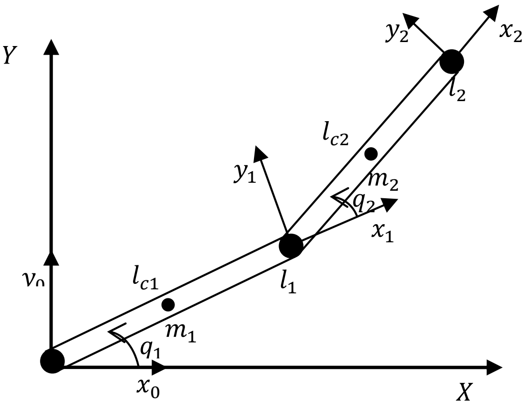

Figure 6. 2-dof robot system

In the simulations, the control aim is to push the joint angles of the two-link robot manipulator to track periodic sinusoidal trajectories.

$$q_d = [2\sin(1.25t) \quad 1.5\ sin\ (1.25t)]^T$$

Table 1. Parameters values

The parameter	Symbol	Value	Unit
link1 length	l_1	0.205	m
link2 length	l_2	0.210	m
link1 center of gravity	l_{c1}	0.1548	m
link2 center of gravity	l_{c2}	0.105	m
link1 mass	m_1	3.55	Kg
link2 mass	m_2	0.75	Kg
link1 inertia	I_1	3.7× 10^{-5}	Kgm2
link2 inertia	I_2	1.47× 10^{-4}	Kgm2
gravity acceleration	g	9.81	ms−2

The parameters of GA are specified in Table 2:

Table 2. Genetic algorithm parameters

Population size	80
Generations Number	120
Crossover Probability	0.8
Variables number	4
Function of selection	Roulette wheel mechanism

The numerical values of the parameters are obtained after a number of experiments.

To validate the viability of the proposed scheme controller, two simulation cases are presented: the first one in absence of external disturbance, and the second one with the external disturbance. The found parameters of the proposed controller are optimised and given in the Table 3, in both cases.

The obtained parameter values are given in Table 3:

Table 3. The optimal values of parameters

K_{11}	K_{12}	K_{21}	K_{22}
102.53	601.35	103	610.22

Case 1: disturbance-free, with initial conditions different of zero

The system initial state value is given by:

$$x_1(0) = [0.2, 0.1] \text{ and } x_2(0) = [0, 0]$$

In this case, the Figures 7 to 11 show the different simulation results.

According to the Figures 7-11, the suggested scheme is skilled to control the two dof robot manipulator. The responses from both links have followed the desired trajectories for the acquired parameters, despite the fact that the starting conditions for the tracked trajectories' initial points differ, and as can be seen, both errors converge to zero quickly.

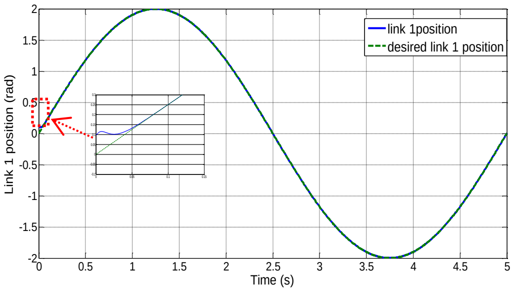

Figure 7. First link's $q_1$angle position response under the initial condition 0.2

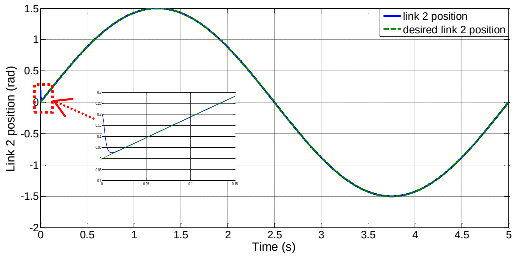

Figure 8. Second link's q_2 angle position response under the initial condition 0.1

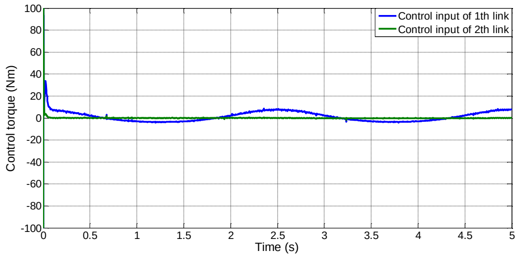

Figure 9. Control torque of both links with initial condition [0.2, 0.1]

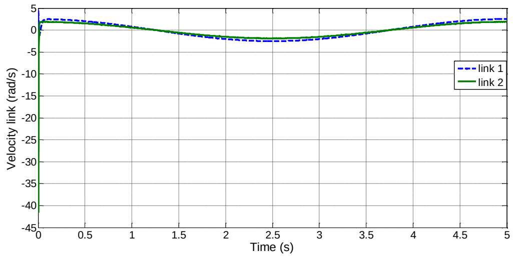

Figure 10. Time responses velocity of both links with initial condition [0.2, 0.1]

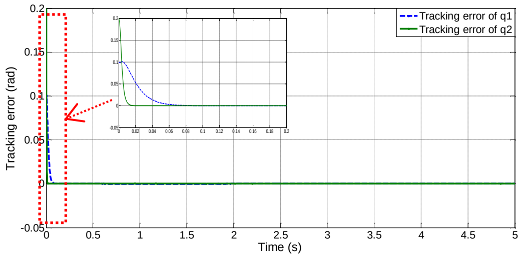

Figure 11. Time responses of tracking errors for both links with initial condition [0.2, 0.1]

The Genetic Algorithm tuned Backstepping controller reveals short convergence time. Through figures 7 and 8 we notice that even when the initial conditions are different from zeros [0.2, 0.1], the system's response joins the reference trajectory in a very short time.

To record the control performance, the position-tracking response's mean square error (MSE) measurement is well described as follows

$$fitness = MSE = \frac{1}{T}\sum_{m=1}^{T}(q_i(m) - q_{id}(m)\ )^2 \qquad (28)$$

Since each of the four aforementioned criteria has benefits and drawbacks, we tried them in our work to determine which is the most effective and applicable to our application. The MSE criterion is very satisfactory in terms of results quality and simulation time.

The associated tracking errors are depicted in Figure 11, the respective MSE values of the tracking errors are 1.8143×10^{-04} and 4.0679×10^{-04}. The convergence of all internal tracking errors (ε_1 and ε_2) in the proposed genetic algorithm based Backstepping control system can be guaranteed according to the stability analysis.

Figure 9 shown that a small chattering control effort exist.

The algorithm's choice of 120 iterations, an 80 individual population size, a crossover operator of 0.8, and a mutation operator of 0.01 is justified in order to produce a population with a high degree of diversity as a result of these reproduction operations. Simple crossover operator and binary mutation are used, along with a roulette wheel selection mechanism, to complete chromosomes reproduction.

The simulation results show that this optimization approach is unable to produce a precise solution for a population of less than 20 chromosomes. To achieve a better solution, the population must contain at least 20 to 30 chromosomes. Moreover, it is noted that using a population with an enormous size (around 200 to 300 chromosomes) had no effect on the fitness function values' progression. On the other hand, as the population increases, the amount of time and memory needed to process data increases as well, which might make large-scale experiments difficult. As a result, we can state that the best population size in terms of the fitness function value and the necessary computational resources is between 80 and 100 people.

Case 2: taking in account the disturbance effect

In this case, we consider the following disturbance, i.e.

$$d = \left[1.5\sin\left(\frac{2\times\pi}{0.2}t\right)\ \ 0.5\,sin\,(\frac{2\times\pi}{0.2}t)\right]^T$$

The disturbance situation is occurring at 2 s.

The results of simulation in this case are presented in Figures 12 to 16.

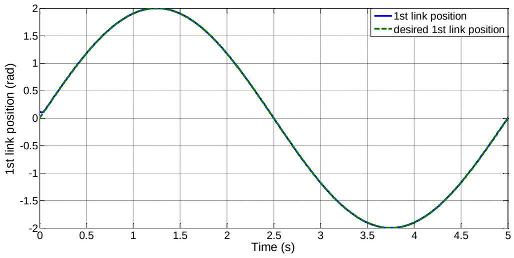

Figure 12. First link's q_1 angle position response under the initial condition 0.2 and disturbance d

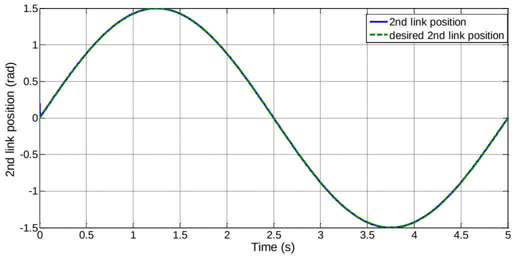

Figure 13. Second link's q_2 angle position response under the initial condition 0.1 and disturbance d

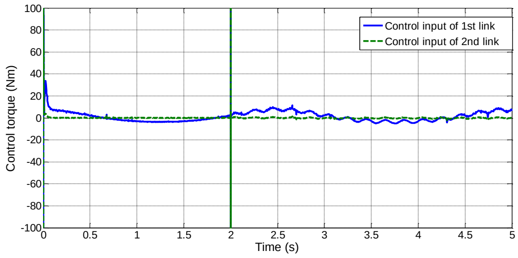

Figure 14. Control torque of both links with initial condition [0.2, 0.1], and disturbance d

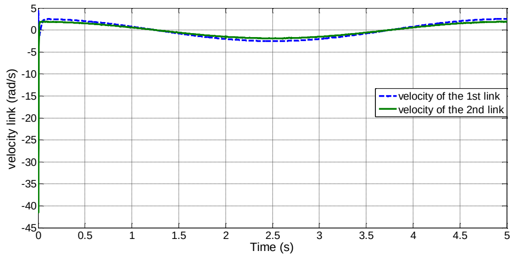

Figure 15. Time responses velocity of both links with initial condition [0.2, 0.1], and disturbance d

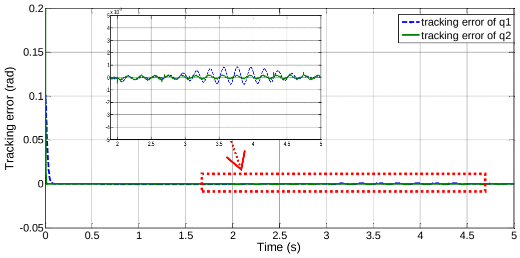

Figure 16. Time responses of tracking errors for both links with initial condition [0.2, 0.1], and disturbance d

At first glance from Figures 12 and 13, we can find that despite the sinusoid disturbance the system will stay still stable, as well as both links' responses tracked the desired trajectories.

By observing Figure 14 serious chattering control torques can be seen, mostly after disturbance occurring.

Figure 15 shows time responses velocity of the two links under disturbance effect.

In Figure 16, it can be seen that the tracking error comes out when the sinusoidal disturbance is introduced. However, the maximum amplitude of the tracking error is as minor as 0.001, and which decreases. As can be observed, link 1 is more susceptible to the effects of disturbances since it bears the stress applied from the second link.

The respective MSE values in this case of the tracking errors in Figure 16 are 1.8664×10^{-04} and 4.0647×10^{-04}.

7. CONCLUSIONS

The present paper investigates a new hybrid control method designed for a class of nonlinear system, which combines two approaches: the backstepping control technique and genetic algorithm.

Backstepping controller is a recursive procedure that divides a design problem for the complete non-linear system into a sequence of problems for reduced order systems; this allows to have in a constructive way a control law allowing to ensure the stability in closed loop of the system using the function of Lyapunov.

In this control scheme, backstepping controller parameters are tuned by using GA optimization method for a two dof robot manipulator control. The main advantages of this method is the fast converging of tracking errors to zero when the initial conditions are different of zero, the stability and robustness against disturbances.

Additionally, where the fitness function for GA is based on error, genetic algorithms are employed to determine the optimal Backstepping control parameters. On other hand, several difficulties might arise when utilizing genetic algorithms, particularly when choosing the population size and generation number to be employed, which often impacts the results quality and the speed of the execution.

Simulation results show that, the algorithm's execution time increases when the population size is increased, necessitating the use of a powerful calculator. On the other hand, if we use a value less than 20, we get poor results, indicating that we haven't reached a global optimum.

We found that the proposed scheme provides a good results trajectory tracking of joint positions and velocities, and attenuates the external disturbance, which was applied on positions, so a fast converging of tracking errors to zero. These results revealed the suggested control schemes effectiveness for the 2-dof robot manipulator.

In future work, other evolutionary optimization techniques may also be employed for the control parameter optimization. A comparison of the results produced using various fitness functions may be the topic of another work.

ACKNOWLEDGMENT

This work was supported by the Direction Générale de la Recherche Scientifique et du Développement Technologique (DG RSDT).

REFERENCES

[1] Wai, R.J., Muthusamy, R. (2012). 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. https://doi:10.1109/tnnls.2012.2228230

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

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

[4] Wei, Y., Zhang, J., Hou, L., Jia, F., Chang, Q. (2013). Backstepping Adaptive Fuzzy Control for two-link robot manipulator. International Journal of Computer Science Issues (IJCSI), 10(1): 303-308.

[5] Krstić, M., Kanellakopoulos, I., Kokotović, P. (1995). Nonlinear and Adaptive Control Design. New York, Wiley.

[6] Salman, M.A., Kadhim, S.K. (2022). Optimal backstepping controller design for prosthetic knee joint. Journal Européen des Systèmes Automatisés, 55(1): 49-59. https://doi.org/10.18280/jesa.550105

[7] Kwan, C., Lewis, F.L. (2000). Robust backstepping control of nonlinear systems using neural networks.

IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 30(6): 753-766. http://dx.doi.org/10.1109/3468.895898

[8] Mbede, J.B., Mvogo Ahanda, J.J.B. (2014). Exponential tracking control using backstepping approach for voltage-based control of a flexible joint electrically driven robot. Journal of Robotics, 2014: 1-10. http://dx.doi.org/10.1155/2014/241548

[9] Ahanda, J.J.B.M., Mbede, J.B., Melingui, A., Essimbi, B. (2018). Robust adaptive control for robot manipulators: Support vector regression-based command filtered adaptive backstepping approach. Robotica, 36(4): 516-534. https://doi.org/10.1017/S0263574717000534

[10] Ahanda, J.J.B.M., Mbede, J.B., Melingui, A., Essimbi, B., Lakhal, O., Merzouki, R. (2017). Robust control for robot manipulators: Support vector regression based command filtered adaptive backstepping approach. International Federation of Automatic Control (IFAC)-PapersOnLine, 50(1): 8208-8213. http://dx.doi.org/10.1016/j.ifacol.2017.08.1385

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

[12] Saidi, K., Allad, M. (2015). Fuzzy controller parameters optimization by using genetic algorithm for the control of inverted pendulum. In Proceedings of the 3rd International Conference on Control, Engineering and Information Technology (CEIT), Tlemcen, Algeria, 25-27 May 2015. http://dx.doi.org/10.1109/CEIT.2015.7233020

[13] Saidi, K., Boumediene, A., Massoum, S. (2020). An optimal PSO-based sliding-mode control scheme for the robot manipulator. Elektrotehniski Vestnik, 87(1/2): 53-59.

[14] VIDYASAGAR, M. (1993). Nonlinear systems analysis, Second Edition, USA: Prentice-Hall International.

[15] Mahmood, T.S., Lutfy, O.F. (2022). A wavelet neural network-based NARMA-L2 feedforward controller using genetic algorithms to control nonlinear systems. Journal Européen des Systèmes Automatisés, 55(4): 439-447. https://doi.org/10.18280/jesa.550402

[16] Roberge, V., Tarbouchi, M., 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. http://dx.doi.org/10.1109/TPEL.2014.2311737

[17] Sivanandam, S.N., Deepa, S.N. (2008). Genetic algorithms. In Introduction to Genetic Algorithms (pp. 15-37). Springer, Berlin, Heidelberg.

[18] Reboucas Filho, P.P., da Silva, S.P.P., Praxedes, V.N., Hemanth, J., de Albuquerque, V.H.C. (2019). Control of singularity trajectory tracking for robotic manipulator by genetic algorithms. Journal of Computational Science, 30: 55-64. https://doi.org/10.1016/j.jocs.2018.11.006

[19] Abdullah, A.I., Alnema, Y.H.S., Thanoon, M.A. (2022). Stabilization of three links inverted pendulum with cart based on genetic LQR approach. Journal Européen des Systèmes Automatisés, 55(1): 125-130. https://doi.org/10.18280/jesa.550113

[20] Sahraoui, K., Kouzi, K., Ameur, A. (2017). Optimization of MRAS based speed estimation for speed sensorless control of DSIM via genetic algorithm. Electrotehnica, Electronica, Automatica, 65(3).

[21] Ar, Y., Bostanci, E. (2016). A genetic algorithm solution to the collaborative filtering problem. Expert Systems with Applications, 61: 122-128. http://dx.doi.org/10.1016/j.eswa.2016.05.021

[22] Fleming, P.J., Purshouse, R.C. (2002). Evolutionary algorithms in control systems engineering: A survey. Control Engineering Practice, 10(11): 1223-1241. https://doi.org/10.1016/S0967-0661(02)00081-3

[23] Irshad, M., Khalid, S., Hussain, M.Z., Sarfraz, M. (2016). Outline capturing using rational functions with the help of genetic algorithm. Applied Mathematics and Computation, 274: 661-678. https://doi.org/10.1016/j.amc.2015.10.014

[24] Ravi, S., Rao, D.N., Krishna, K.M., Mezhuyev, V. (2014). A proposed GA based PID controller for three phase brushless DC motor. Universal Journal of Control and Automation, 2(4): 69-76. http://dx.doi.org/10.13189/ujca.2014.020401

[25] Amini, Y., Gerdroodbary, M.B., Pishvaie, M.R., Moradi, R., Monfared, S.M. (2016). Optimal control of batch cooling crystallizers by using genetic algorithm. Case Studies in Thermal Engineering, 8: 300-310. http://dx.doi.org/10.1016/j.csite.2016.09.001

[26] Dombre, E., Khalil, W. (Eds.). (2007). Modeling, performance analysis and control of robot manipulators. Washington: Wiley Online Library http://dx.doi.org/10.1002/9780470612286

NOMENCLATURE

GA	Genetic Algorithm
BSC	Backstepping Control
DoF	Degrees of Freedom
MIMO	Multiple-Input–Multiple-Output
PSO	Particle Swarm Optimization
WOA	Whale Optimization Algorithm
ACO	Ant Colony Optimization
ROA	Rider Optimization Algorithm