Terminal and Backstepping Sliding Mode Control with Genetic Algorithms for Robot Manipulators

Umut TİLKİ*, Melikcan ÖLGÜN

Suleyman Demirel University, Faculty of Engineering, Electrical and Electronics Engineering Department, Isparta, 32260, Turkey

umuttilki@sdu.edu.tr (**Corresponding author*), melikcanolgun@gmail.com

Abstract: This paper presents two controllers that address the negative effects of external disturbances, parameter variations, and random noise on trajectory tracking for robot manipulators. These controllers were designed for handling variable-mass loads, uncertainties, and disturbances in the robot's dynamic equations. In order to achieve an accurate trajectory tracking and an improved disturbance rejection, Terminal Sliding Mode Control (TSMC) was employed in combination with a Neural Network (NN) with radial basis functions that modelled the unidentified system characteristics. The employed control algorithms, namely the Neural Network-Based Terminal Sliding Mode Control (NNBTSMC) and the Neural Network-Based Backstepping Terminal Sliding Mode Control (NNBBTSMC), were optimized using genetic algorithms in order to determine the optimal controller coefficients. Simulation tests were conducted on a two-link robot manipulator, and the effectiveness of the proposed controllers was demonstrated. Additionally, a comparison was made between the analysed controllers regarding their trajectory tracking performance and control inputs for the case when the manipulator payload changed over time. The simulation results showed that the NNBBTSMCWGA outperformed the traditional SMC and NNTSMC by exhibiting a higher robustness, and reduced chattering effects on the control inputs.

Keywords: Robot manipulator control, Sliding mode control, Robust control, Radial basis function networks.

1. Introduction

Robotics research now focuses on controlling the trajectory of manipulator systems due to the rapid advancement of robot technology and its widespread application. In addition to their typical industrial uses, robot manipulators are utilized for highly precise tracking tasks such as space and underwater exploration, as well as for remote surgical operations. As the demand for high performance and precision increases, including rapid response and accurate trajectory tracking, new control techniques for robot manipulators are being developed. However, these manipulators are complex systems with nonlinear relationships between position, velocity, acceleration, and torques applied to their joints. Moreover, constructing an ideal controller is often unfeasible due to the nonlinear dynamics of the system, model uncertainties, coupling, frictional effects, and elasticity of manipulator linkages. The primary objective of trajectory tracking is to improve system stability while minimizing trajectory tracking errors, which represent the differences between the reference input and the actual output (Lewis, Dawson & Abdallah, 2003).

In this particular research, two types of control techniques, namely Neural Network-Based Terminal Sliding Mode Controller (NNTSMC) and Neural Network-Based Backstepping Terminal Sliding Mode Controller (NNTSMC), are evaluated in order to determine their effectiveness as optimal controllers. Furthermore, their performances are enhanced by utilizing a genetic algorithm.

The fundamental concept behind the developed controllers is Sliding Mode Control (SMC), which can be divided into two parts. The first part involves creating appropriate sliding surfaces that determine the trajectory of the state variables to be followed. The second part brings the state variables to these predetermined sliding surfaces by switching robust control laws created in the first part (Vijay & Jena, 2018). Conventional Sliding Mode Control (CSMC) achieves asymptotic convergence of the state due to the linearity of the switching plane (Amer, Sallam & Elawady, 2011). However, this convergence can only be achieved in an infinite amount of time, although it can be expedited by adjusting the SMC parameters. By contrast, Terminal Sliding Mode Control (TSMC) utilizes a nonlinear function on the sliding surface, which provides high precision, robustness, and rapid convergence within a finite time despite uncertainties (Tilki & Erüst, 2021; Nguyen et al., 2018). Nevertheless, the chattering phenomenon represents a significant drawback of TSMC, as high-frequency oscillations caused by discontinuities in control signals result in poor control accuracy, excessive wear of mechanical parts, and potential damage to robot joints.

To improve the position control of a compliant rescue manipulator actuated by a tendon sheath, neural network-based algorithms have been

https://doi.org/10.24846/v32i2y202311

widely employed in model-free approaches. Wu et al. (2019) proposed a sliding mode control technique based on neural networks to enhance the position control of the manipulator's gripper, despite modelling uncertainties and external disturbances. In this paper, a RBF network was proposed to achieve precise position control, and the NN-based SMC controller and radial basis function weights were adjusted in real time to counteract chattering and disturbances. Furthermore, prior knowledge of the actual robot system was not required, which made it an effective method for enhancing the performance of tendon sheath-actuated manipulators. Ren, Wang & Chen (2020) proposed similar approaches in their work. They mitigated the undesirable effects of chattering caused by high-frequency switching terms in the first derivative of the control signal in SMC by employing a continuous control law (Yi & Zhai, 2019). Model-free sliding mode control algorithms, which do not require a mathematical model of the system being controlled, are particularly useful in situations where obtaining an accurate model is difficult or impractical (Precup et al., 2017; Zhu, 2021). These algorithms are primarily designed to handle the uncertainties and disturbances inherent to nonlinear systems, making them robust and adaptable. To address the singularity problem associated with TSMC, they proposed a power-reaching law and utilized the inverse of the tangent function to eliminate singularities without requiring auxiliary measures. Consequently, the tracking error converged to the origin from any initial state within a finite time (Zhai & Xu, 2020). Another model-free approach integrated a terminal sliding surface and an observer to ensure continuous sliding mode tracking (Zhang et al., 2018). For the control of a single-link rigid robot manipulator with an unknown constant payload, Li, Wang & Yu (2021) combined a sliding mode control approach with a nonlinear disturbance observer. The experimental results demonstrated the effectiveness of the observation of the unknown payload handling. Truong, Vo, and Kang (2021) employed a backstepping global fast terminal sliding mode control strategy for trajectory tracking control in industrial robotic manipulators, improving dynamic performance of the manipulator by utilizing an integral of the global fast terminal sliding mode surface and achieving finite-time convergence in both SMC and TSMC for rapid convergence.

Numerical simulations were conducted on a two-link robot manipulator by Norsahperi & Danapalasingam (2019) to examine the efficiency of various control approaches. They utilized trajectory tracking and energy consumption as performance measures. The tested controllers included integral sliding mode control (ISMC), linear-quadratic regulator with integral action (LQRT), and optimal integral sliding mode control (OISMC). Genetic algorithms were employed to obtain the optimal control settings and solve optimization problems for the proposed linear quadratic regulator approaches.

Terminal Sliding Mode Control (TSMC) with nonlinear sliding surfaces results in chattering effects that can be minimized through Backstepping Terminal Sliding Mode Control (BTSMC). This paper proposes a new method called Neural Network-Based Backstepping Terminal Sliding Mode Control with Genetic Algorithm (NNBBTSMCWGA). In this approach, control inputs are generated based on the outputs of a neural network. However, since the neural network's approximation formula introduces some error, additional uncertainty is added to the system during the generation of control inputs (Sun et al., 2011). An adaptive control approach using a terminal sliding mode controller, which is robust against uncertainties and noise in neural network-based systems, was employed. The authors achieved robust trajectory tracking against model uncertainties and artificially added noise using these control structures.

The main contributions of this paper are as follows:

1. The sliding mode-based control algorithms are enhanced by using a genetic algorithm to determine the coefficients of the terminal sliding mode controller and artificial neural network. This combination of neural network-based models with a genetic algorithm is a widely used method for optimizing control structures, as it was demonstrated by previous studies (Boukadida, Benamor & Messaoud, 2019; Zhang, Zhuang & Du, 2009). The genetic algorithm is employed to determine controller coefficients that enable the manipulator system to achieve an optimal trajectory tracking behavior. Additionally, the genetic algorithm eliminates the problem of determining controlled coefficients in complex systems. Therefore, an increase in the number of unknown coefficients resulting from changes in the control structure of the system does not lead to significant issues;

2. The algorithms proposed in this paper ensure that the tracking error will converge to zero within a finite time. The Lyapunov Theory is utilized to establish the asymptotic stability of the closed-loop system, thereby improving the manipulator's trajectory tracking accuracy;

3. Numerical simulations were conducted on a 2-degree-of-freedom robot manipulator using the Matlab/Simulink environment to demonstrate the effectiveness of the proposed controller. The experimental results indicate that the proposed controller is not only fast but also possesses disturbance rejection and robustness properties. The enhanced NNBBTSMCWGA provides control inputs that are free from chattering.

This paper is organized as follows: Section 2 presents the mathematical basis for controlling the robot manipulator's trajectory; Section 3 introduces the control approaches that have been modified by using a genetic algorithm. In Section 4, simulation results for these controllers are presented, along with a comparison of these results. Finally, Section 5 provides the concluding remarks.

2. Manipulator Dynamics

A robot manipulator is an electronically controlled mechanical mechanism composed of multiple components. The operation of robot manipulators is controlled by determining the positions and directions of the various joints that generate the motion of the end effector. In this paper, the control of a robot manipulator's trajectory while it is subject to uncertainties and outside disturbances is taken into consideration.

In this work, a sliding mode controller structure based on neural networks whose coefficients are determined by a genetic algorithm is proposed for the calculation of torque values applied to each joint of the robot manipulator for position control. Since this technique is based on the calculation of the joint torques, it is called computational torque control. Generally, there are two modelling approaches for robot manipulators, which are the Lagrange-based and Newton-Euler methods. In this study, the manipulator dynamics is derived by using Lagrangian mechanics. In the framework of this method, the required control torques to be applied to each joint are determined by joint variables, which are the joint angles. According to the Lagrangian mechanics, the equation of motion for a n-link robot manipulator is given in equation (1).

$$M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau \quad (1)$$

Position, velocity, and acceleration are represented in this equation by $q, \dot{q}, \ddot{q} \in R^n$ respectively. Moreover, $M(q) \in R^{nxn}$ expresses a symmetric inertial matrix that is positive definite. $V(q,\dot{q}) \in R^{nxn}$ represents the coriolis and centripetal force matrix. $G(q) \in R^{nx1}$ represents the gravity component in the torque equation. As it was previously indicated, the unknown model parameters leave the entire system susceptible to external disturbances and uncertainty (friction). In the equation of motion, these terms are represented by $F(\dot{q})$ and τ_d. Equation (2) is a reordering of the equation of motion (equation (1)).

$$\ddot{q} = M^{-1}(q)[\tau - (V_m(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d)] \quad (2)$$

The values of the matrices and vectors, as well as the units utilized in the above-mentioned equations for the robot manipulator are expressed as follows:

$$M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \quad (3)$$

$$V_m(q,\dot{q}) = \begin{bmatrix} V_{m11} & V_{m12} \\ V_{m21} & V_{m22} \end{bmatrix} \quad (4)$$

$$M_{11} = (m_1 + m_2)a_1^2 + m_2 a_2^2 + 2m_2 a_1 a_2 \cos\theta_2 \quad (5)$$

$$M_{12} = M_{21} = m_2 a_2^2 + m_2 a_1 a_2 \cos\theta_2 \quad (6)$$

$$M_{22} = m_2 a_2^2 \quad (7)$$

$$V_{m11} = -m_2 a_1 a_2 \dot{\theta}_2 \sin\theta_2 \quad (8)$$

$$V_{m12} = -m_2 a_1 a_2 (\dot{\theta}_1 + \dot{\theta}_2) \sin\theta_2 \quad (9)$$

$$V_{m21} = m_2 a_1 a_2 \dot{\theta}_1 \sin\theta_2, \; V_{m22} = 0 \quad (10)$$

$$G(q) = \begin{bmatrix} (m_1 + m_2) g a_1 \cos\theta_1 + m_2 g a_2 \cos(\theta_1 + \theta_2) \\ m_2 g a_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (11)$$

$$F(\dot{q}) = \begin{bmatrix} \dot{\theta}_1 + 2\,\mathrm{sgn}(\dot{\theta}_1) \\ \dot{\theta}_2 + 2\,\mathrm{sgn}(\dot{\theta}_2) \end{bmatrix} \quad (12)$$

An external disturbance is modelled as a random noise with a magnitude bound. To test and validate the suggested control strategies, a two-link robot manipulator is utilized. The axis representation of the robot manipulator, for which the proposed control algorithms will be validated, is given in Figure 1. The variable angles are denoted as θ_1 and θ_2, while a_1 and a_2 represent the link lengths and m_1 and m_2 represent the link weights. In this paper, the weights of the links are considered to vary with time in order to increase the uncertainty of the analysed system. The numerical values of the link weights and lengths are given in Section 4. The main objective of this paper is to demonstrate the performance of the proposed controller using a genetic algorithm. This allows the manipulator to robustly track the reference trajectory in the presence of high uncertainty and random noise.

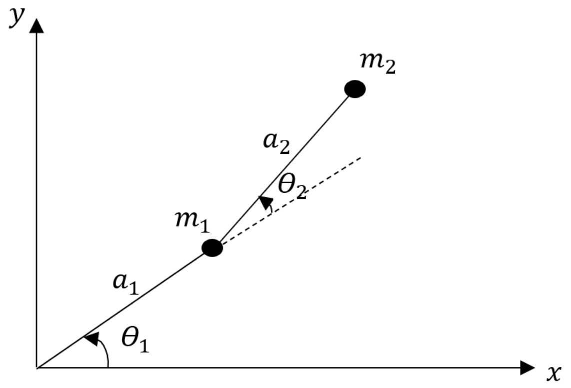

Figure 1. Two-link robot manipulator

3. Control Methods

This section introduces two control strategies for the robot manipulator system with nonlinear dynamics. These techniques rely on backstepping terminal sliding mode control and terminal sliding mode control approaches. Radial basis function neural networks (RBFNN) are utilized in both control techniques to approximate the elements $M(q)$, $V_m(q,\dot{q})$, and $G(q)$. Consequently, the RBFNNs will approximate the dynamic equation. A finite-time control mechanism with a suitable updating law is constructed to force the system states to the sliding surface and to converge to zero in a finite amount of time. Furthermore, the genetic algorithm is utilized to compute the coefficients of the controllers and artificial neural networks (ANNs). The determination procedure is critical since the value of control coefficients directly affects the system's performance and overall stability.

3.1. Terminal Sliding Mode Control Structure

Artificial neural networks (ANNs) are employed to approximate the unknown nonlinearities in a system, as it was mentioned in the adaptive control literature (Tran & Kang, 2017). The neural network function is illustrated by the following equation:

$$f_1(x) = W^T \phi(X) + \varepsilon(X) \tag{13}$$

In this equation, $W \in R^{nX2}$ represents the weight matrices and X represents the input vector. Each component of W represents the coefficient of the ϕ function. $\varepsilon(X)$ denotes the neural network's approximation error. ϕ is constructed as a vector with n>1 representing the number of neurons, and it is represented as follows:

$$\phi(X) = [\phi_1(X) \quad \phi_2(X) \quad \dots \quad \phi_n(X)]^T \tag{14}$$

where $\phi_i(X)$ given in equation (15) is the RBF:

$$\phi_i(X) = e^{\left(\frac{\|x-c_i\|^2}{\sigma_i^2}\right)} \quad i = 1,2,\dots,n \tag{15}$$

Here, c_i and σ_i refer to the center and width of the neuron, respectively. Since a nonlinear sliding surface is used in the terminal sliding mode control structure, the robot manipulator is able to reach the desired reference trajectory faster than with a conventional sliding-mode control. As a result, the error functions of the manipulator converge to zero in a finite time with high precision and strong stability.

The sliding surface used in this controller structure is expressed in equation (16).

$$s = \dot{e} + \Lambda |e|^{\varphi} \operatorname{sgn}(e) \tag{16}$$

The corresponding control rule is employed for maintaining the trajectory of the system state on the sliding surface as soon as the system state reaches it. The continuous control law (equation (17)) obtained by setting the sliding surface equation to $\dot{s} = 0$ for a nominal system in the absence of uncertainties and external disturbances, as it was explained in the paper of Tran & Kang (2017), can be considered as the equivalent control law.

$$\dot{s} = \ddot{e} + \varphi \Lambda |e|^{\varphi - 1} \dot{e} \tag{17}$$

Φ and Λ are positive constants between 0 and 1 in equations (16) and (17). Equation (18) will be further approximated by the RBF.

$$\begin{aligned} M(q)\dot{s} &= M(\ddot{q}_d - \ddot{q} + \varphi \Lambda |e|^{\varphi-1} \dot{e}) \\ &= M(\ddot{q}_d + \varphi \Lambda |e|^{\varphi-1} \dot{e}) - M\ddot{q} = 0 \end{aligned} \tag{18}$$

$$M(q)\dot{s} = M(\ddot{q}_d + \varphi\Lambda |e|^{\varphi-1}\dot{e}) - V_m s + V_m(\dot{q}_d + \varphi\Lambda |e|^{\varphi-1}\dot{e}) + G + F + \tau_d - \tau \quad (19)$$

The RBF will eventually approximate the $f(X)$ function that is derived from equation (20).

$$f(X) = M(q)(\ddot{q}_d + \varphi\Lambda |e|^{\varphi-1}\dot{e}) + V_m(\dot{q}_d + \varphi\Lambda |e|^{\varphi-1}\dot{e}) + G + F \quad (20)$$

The following controller law is formulated by assuming that the sliding variable will converge to zero eventually when introduced into equation (21):

$$\tau = f(X) - V_m s + \tau_d \quad (21)$$

By taking into account both equation (21) and the error function from the RBF, the designed controller law takes the following form:

$$\tau = f_1(X) + K_v s + v \quad (22)$$

In this case, to make the system robust, K_v is used as a controller coefficient. To eliminate both the disturbing torque τ_d and the RBF-induced approximation error $\varepsilon(X)$, the control rule for this type of controller incorporates the variable v:

$$v = b\,\mathrm{sgn}(s) \quad (23)$$

In equation (23), *sgn* represents the signum function, and b is the controller coefficient. In equation (23) v will provide resistance against noise and error since $f_1(X)$ will converge to $f(X)$. K_v will guarantee the system's stability. Equation (20) can be used to find the inputs of the $f_1(X)$ function as $X = [\ddot{q}_d \quad \dot{e} \quad \dot{q}_d \quad e \quad q_d]$.

Theorem 1: Consider the manipulator modelled by equation (1). The control input of the system is given by equation (22). The weight matrices of the neural network satisfy the given condition $\dot{\tilde{W}} = -F_w \phi s^T$. The trajectory tracking errors e and $\dot{e}$ will asymptotically converge to zero if the controller coefficient K_v is positive.

Proof: The positive definite Lyapunov function that was chosen for this controller is expressed in equation (24). K_v is shown to be positive by the Lyapunov function.

$$V = \frac{1}{2}s^T M s + \frac{1}{2}tr(\tilde{W}^T F_w^{-1}\tilde{W}) \quad (24)$$

In this equation, tr refers to "tracing", $\tilde{W} = W^* - W$, W^* is ideal weights matrix. F_w is a positive definite matrix. By differentiating both sides of the Lyapunov function and substituting $M(q)\dot{s}$ term into the equation for $\dot{V}$ one obtains:

$$\dot{V} = -s^T K_v s + \frac{1}{2}s^T(\dot{M} - 2V_m)s + tr\tilde{W}^T(F_w^{-1}\dot{\tilde{W}} + \phi s^T) - s^T(v - \varepsilon + \tau_d) \quad (25)$$

Since $\dot{M} - 2V_m(q,\dot{q})$ is a skew symmetric matrix, the second term in equation (25) becomes zero (i.e. $\frac{1}{2}s^T(\dot{M} - 2V_m)s = 0$). If $\dot{\tilde{W}} = -F_w \phi s^T$ is selected, the equation becomes:

$$\dot{V} = -s^T K_v s - s^T(v - \varepsilon + \tau_d) \le 0 \quad (26)$$

It can be concluded that if $\dot{s} = 0$, then $\dot{V} = 0$. Thus, the manipulator trajectory is guaranteed to achieve sliding manifold $s = 0$ if the controller coefficient K_v is positive. This demonstrates the system's asymptotic stability, so the tracking errors of position and velocity (e and $\dot{e}$) of the system asymptotically converge to zero as t goes to infinity.

3.2 Backstepping Terminal Sliding Mode Control Structure

For this controller structure, it was decided to change the sliding surface due to the occurrence of chattering phenomena on the sliding surface for the terminal sliding-mode controller caused by too many sharp transitions. The controller described in this subsection is aimed to achieve rapid reaching time while maintaining a similar balance between velocity and minimum trajectory tracking errors. Additionally, this controller is meant to reduce the chattering effects that occur in the control signal. For the controller design as adaptive backstepping terminal sliding mode control (ABTSMC), equation (21) is used, and the Lyapunov stability theorem is applied as in equation (24) from the previous subsection (Yen, Nan & Cuong, 2019). A nonlinear sliding surface has been chosen to ensure that the sliding surface reaches zero in a finite time, as it is expressed in equation (27).

$$s = \dot{e} + \frac{1}{\Lambda}|e|^{\varphi} + (1+\lambda)e \quad (27)$$

In this equation, Λ, λ and φ are positive constants. All of these controller coefficients are determined using a genetic algorithm.

3.3 Genetic Algorithm for Control Structure

The genetic algorithm is a computational method inspired by natural evolution. It helps one to find optimal solutions to complex problems by exploring a vast search space. In this case, the goal is to minimize the difference between the desired trajectory and the actual trajectory of the system being controlled. To achieve this, the genetic algorithm takes trajectory-tracking errors as inputs and generates the controller coefficients and neural network parameters as outputs. It approximates the center (c_i) and width (σ_i) of the neurons in the ANN, which then determine the weight coefficients (W) of the neural network. Additionally, the genetic algorithm determines the coefficients of the implemented controller (K_v, b, Λ, λ and ϕ) used in conjunction with the TSMC. These coefficients significantly influence the performance of the entire analysed system.

The genetic algorithm operates through a series of iterations. It starts by generating a population that includes all the aforementioned coefficients. The algorithm then goes through a process of selection, crossover, mutation, and evaluation of suitability for each member of that population. This process is guided by a plan outlined in the algorithm. By repeating these iterations, the genetic algorithm gradually converges towards an optimal solution that minimizes the trajectory-tracking error. Once the algorithm completes all possible iterations, the most suitable population of coefficients is selected to be used as the controller coefficients (Zhang, Zhuang & Du, 2009).

Figure 2 depicts a general block diagram of the genetic algorithm-based controller.

Algorithm. GA-Based Controller Design	
Input **Output**	Population size, Max. iteration number Optimal coefficients for the controller
1	InitializePopulation()
2	Set iterationCount = 0.

3	while iterationCount < MaximumIterations
	FitnessEvaluation():
	for i = 1 to PopulationSize:
	Set individual = population[i].
	Calculate the error
	Set fitnessValue = calculateFitness(error)
	Set individual.fitness = fitnessValue.
	Set population[i] = individual.
	Selection():
	Create an empty array selectedPopulation
	for i = 1 to PopulationSize:
	Set individual = population[i].
	if individual.fitness > thresholdFitness:
	Add individual to selectedPopulation.
	Set population = selectedPopulation.
	Crossover():
	Create an empty array newPopulation.
	for i = 1 to PopulationSize/2:
	Set parent1, 2 = randomly select
	Set child1, child2 =
	performCrossover(parent1, parent2)
	Add child1, child2 to newPopulation.
	Set population = newPopulation.
	Mutation():
	for i = 1 to PopulationSize:
	Set individual = population[i].
	for each coefficient in individual's
	coefficients:
	if random() < mutationProbability:
	Modify the coefficient randomly within a
	certain range.
	Set population[i] = individual
	Set iterationCount = iterationCount + 1

4	Select the individual with the highest fitness value
5	return the coefficients of the selected individual.

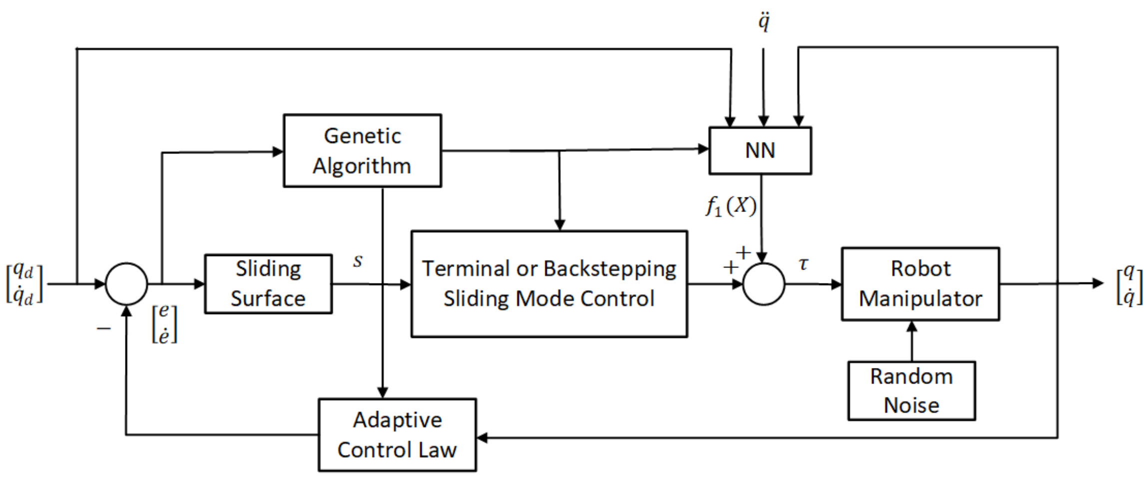

Figure 2. Genetic algorithm-based controller block diagram

4. Numerical Simulations

This study proposes two new controller techniques for improving the trajectory tracking control for a robot manipulator, which is represented using Lagrangian mechanics. The effectiveness and advantages of the suggested controllers are demonstrated using a two-link robot manipulator through numerical simulations in the MATLAB/Simulink environment. Additionally, the steady-state and transient performances of the two proposed controllers, that is the Neural Network-Based Sliding Mode Controller (NNBSMC) and Neural Network Based Sliding Mode Controller with Genetic Algorithm (NNBSMCWGA), are compared for the same desired trajectory, disturbances, and uncertainties. The simulation time is set to 20 seconds. To demonstrate how the controllers are adapted to real-life situations and how the system reacts when there are uncertainties and noise, disturbance terms have been added to the system. One of these uncertainties is represented by the link weights, which are chosen as variables with time. The change in link weights corresponds to payload changes, which cause large variations in the dynamics of the robot.

Figure 3 shows the change in the link weights of the robot manipulator during the simulation, which is used in order to analyse all the employed controllers. Since the weights of the links affect the dynamic equation of the robot manipulator, the robustness of the system's response to variable parameters will be tested through simulation experiments. The fixed parameters for the robot manipulator used in this simulation are included in Table 1.

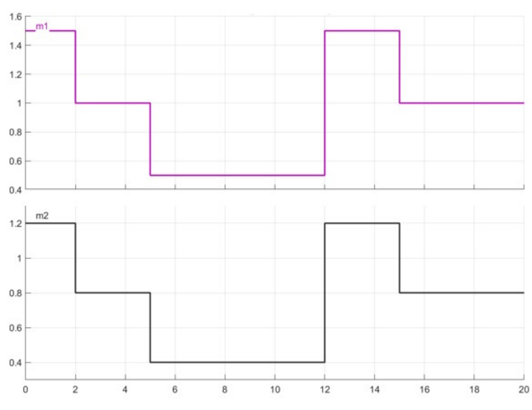

Figure 3. Robot manipulator link weights in relation to time

Table 1. Robot manipulator fixed parameters

Notation	Meaning	Value
a_1	length of link 1	$1m$
a_2	length of link 2	$1m$
g	gravitational acceleration	$9.8\,m/s^2$

The robot manipulator is initially at rest, with:

$$q(0)=\left[q_1(0)\quad q_2(0)\right]^T=0$$

$$\dot{q}(0)=\left[\dot{q}_1(0)\quad \dot{q}_2(0)\right]^T=0$$

as the initial conditions. The disturbance term is a random noise with a bounded magnitude. The simulation parameters are listed in Table 2.

The trajectory tracking performances of the proposed controllers (NNBTSMCWGA and NNBBTSMCWGA) are compared with those of the conventional controllers - neural network-based SMC (NNBSMC) and its improved version with genetic algorithm (NNBSMCWGA) for link 1 and link 2, as it is illustrated in Figure 4.

Table 2. Simulation parameters

Notation	Meaning	Value
q_{d1}	Link 1 trajectory	$\frac{\pi}{2}-\frac{\pi}{4}\cos(0.5t)$
q_{d2}	Link 2 trajectory	$\frac{\pi}{3}\sin(0.5t)$
τ_{d1}	Link 1 noise	$\lvert\tau_{d1}\rvert\le 1$
τ_{d2}	Link 2 noise	$\lvert\tau_{d2}\rvert\le 1$
$F(\dot{\theta}_1)$	Link 1 Friction	$\dot{\theta}_1+2\,\mathrm{sgn}(\dot{\theta}_1)$
$F(\dot{\theta}_2)$	Link 2 Friction	$\dot{\theta}_2+2\,\mathrm{sgn}(\dot{\theta}_2)$

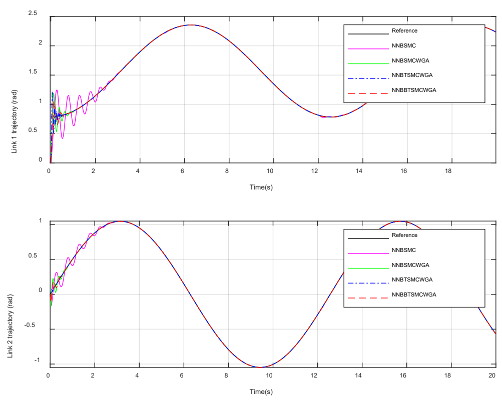

Figure 4. Joint angle trajectory tracking for each link

To analyse the controllers' transient phase, the detailed trajectory tracking performance for each link is shown in Figure 5. The obtained results indicate that the traditional NNBSMC and NNBSMCWGA have slower responses in comparison with the proposed controllers. The proposed controllers reach the desired trajectory in less than 1 second for each link. Additionally, NNBBTSMCWGA exhibits less oscillatory responses than the other controller approaches for both links.

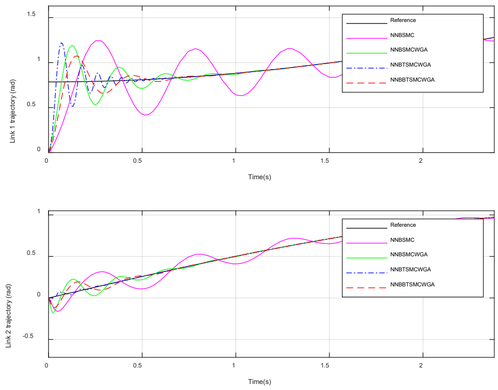

Figure 5. Trajectory tracking for each link in the transient phase

In Figure 6, angular velocity tracking performance for both links is illustrated. As it can be seen, the proposed controllers perform better with regard to angular velocity tracking than the other employed methods. As it can be seen from Figure 6, NNBBTSMCWGA has a less oscillatory response and shows fast convergence, but, on the other hand, for the second link, NNBTSMCWGA has many fluctuations in relation to the reference angular velocity trajectory. The reason for that lies in the use of chattering control input for this controller approach.

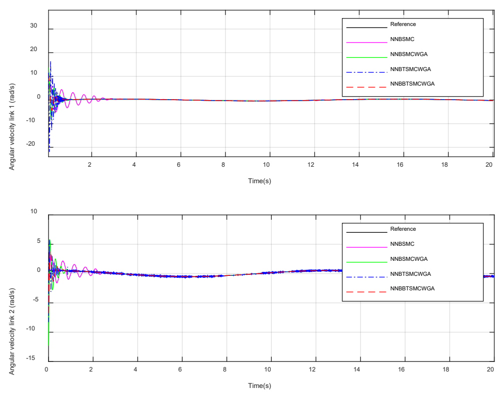

Figure 6. Angular velocity tracking performance

Figure 7 shows the joint angle tracking errors for both links. The error graphs indicate that the traditional NN-based SMC has fluctuations for both links. By contrast, the proposed genetic algorithm-based controllers show a better performance in terms of reaching time and robust response against link weight changes and disturbances.

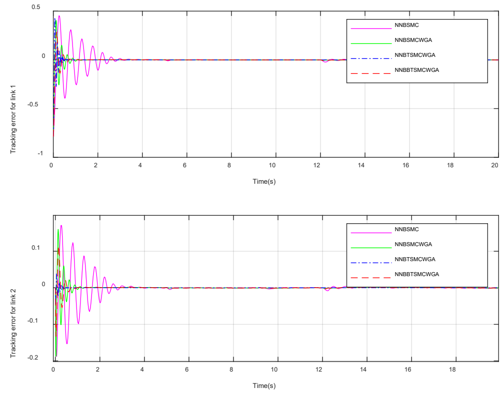

Figure 7. Joint angle tracking errors for both links

The control inputs applied to the joints of the robot manipulator during the simulations are shown in Figure 8. This figure shows that traditional NN-based SMC and its improved version with a genetic algorithm, as well as NNBBTSMCWGA exhibit chattering-free behavior for both links. Although it can be concluded that the trajectory tracking performance of NNBTSMCWGA is the best and the tracking error of each link converges to zero in a minimum time, the control input graphics for this controller (Figure 9) show that it has very large fluctuations. The reason for having these chattering effects lies in the use of the signum function in the torque formula.

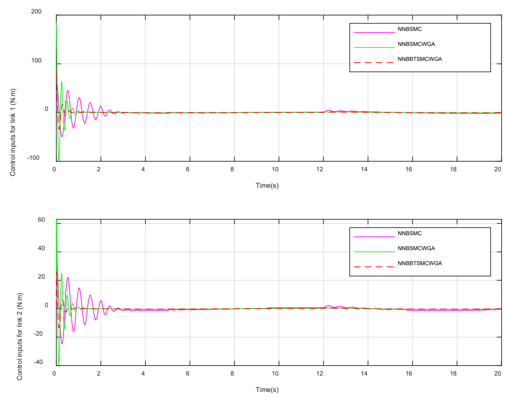

Figure 8. Control input torques

Although it enables a quick switching to a stable state, having excessive chattering in control inputs is a disadvantage for this controller. Robot manipulators are susceptible to chattering effects, which significantly affect their electromechanical systems. This phenomenon can harm robot manipulators in real-time applications by reducing the accuracy of the control system, consuming more energy, and even generating severe vibrations in the system. To mitigate these effects and reduce the number of sharp transitions, the terminal sliding mode controller was improved with backstepping as the second proposed controller structure.

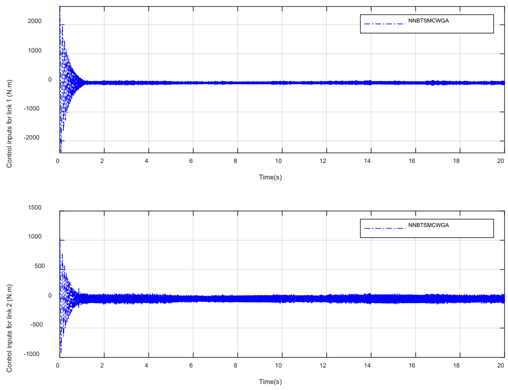

Figure 9. Control input torques for NNBTSMCWGA

A comparison between the above-mentioned controllers was conducted using root mean square (RMS) error during the trajectory tracking performance. The RMS errors ($E_{i_{RMS}}$) were calculated according to the following equation, that is ($E_{i_{RMS}} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\|e_k\|^2}$, $i = 1,2$) where n is the number of simulation steps.

The results of this comparison are shown in Table 3. As it can be seen from Table 3, minimum RMS error values occur when using NNBTSMCWGA. This situation was already predictable when control torques were compared. As a result of the signum function used for NNBTSMCWGA, the control inputs interact with the system, leading to lower RMS errors for the robot manipulator joints.

Table 3. RMS errors for each link in radians

Controller	Link 1 Error	Link 2 Error
NNBSMC	0.1030	0.0258
NNBSMCWGA	0.0746	0.0147
NNBTSMCWGA	0.0133	5.717910^{-4}
NNBBTSMCWGA	0.0610	0.0093

5. Conclusion

In this paper, two new controller techniques are proposed for robotic manipulators to ensure robust trajectory tracking in the face of disturbances and uncertainty. The controller block diagram consists of three sub-sections: a neural network used for predicting unknown model parameters, a sliding mode controller employed for reliable trajectory tracking, and a genetic algorithm utilized for computing the controller parameters and neural network coefficients. The sliding mode controller laws are obtained by using the Lyapunov stability theorem. The use of a genetic algorithm for computing optimum controller parameters leads to a more robust system since the system response is not affected by noise.

Numerical simulations were carried out in the MATLAB/Simulink environment to evaluate the effectiveness of the proposed controller techniques. To increase model uncertainty, link weights were chosen to vary over time. Furthermore, the results obtained for the two proposed controllers were compared with those obtained for the traditional sliding mode-based controllers. The proposed techniques provided reliable trajectory-tracking results with bounded error under disturbances. The NNBTSMCWGA controller exhibited a better performance, both in the transient and steady-state phase, than the NNBBTSMCWGA. However, the main disadvantage of the former controller is the chattering effect on control inputs due to the signum function. On the other hand, the latter controller features no chattering effect, since control torques were determined according to the defined nonlinear sliding surface, and there is no signum term in the control torque equation. Moreover, the transient and steady-state performance of the NNBBTSMCWGA controller is satisfactory. The results showed that NNBBTSMCWGA controller responds rapidly, and with minimal errors, converges to the desired trajectory. Furthermore, the chattering phenomenon is significantly reduced with NNBBTSMCWGA.

REFERENCES

Amer, A. F., Sallam, E. A. & Elawady, W. M. (2011) Adaptive fuzzy sliding mode control using supervisory fuzzy control for 3 DOF planar robot manipulators. *Applied Soft Computing*. 11(8), 4943-4953. doi: 10.1016/j.asoc.2011.06.005.

Boukadida, W., Benamor, A. & Messaoud, H. (2019) Multi-objective design of optimal sliding mode control for trajectory tracking of SCARA robot based on genetic algorithm. *Journal of Dynamic Systems, Measurement, and Control*. 141(3). 031015. doi: 10.1115/1.4041852.

Lewis, F. L., Dawson, D. M. & Abdallah, C.T. (2003) *Robot Manipulator Control: Theory and Practice*. CRC Press.

Li, G., Wang, S. & Yu, Z. (2021) Adaptive nonlinear observer–based sliding mode control of robotic manipulator for handling an unknown payload. In *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*. 235(3), 302-312. doi: 10.1177/0959651820969461.

Nguyen, K., Nguyen, T., Bui, Q. & Pham, M. (2018) Adaptive antisingularity terminal sliding mode control for a robotic arm with model uncertainties and external disturbances. *Turkish Journal of Electrical Engineering and Computer Sciences*. 26(6), 3224-3238. doi: 10.3906/elk-1711-137.

Norsahperi, N. M. H. & Danapalasingam, K. A. (2019) A comparative study of LQR and integral sliding mode control strategies for position tracking control of robotic manipulators. *International Journal of Electrical and Computer Engineering Systems*. 10(2), 73-83.

Precup, R. E., Radac, M. B., Roman, R. C. & Petriu, E. M. (2017) Model-free sliding mode control of nonlinear systems: Algorithms and experiments. *Information Sciences*. 381, 176-192. doi: 10.1016/j.ins.2016.11.026.

Ren, B., Wang, Y. & Chen, J. (2020) Trajectory-tracking-based adaptive neural network sliding mode controller for robot manipulators. *Journal of Computing and Information Science in Engineering*. 20(3), 1-23. doi: 10.1115/1.4047073.

Sun, T, Pei, H., Pan, Y., Zhou, H. & Zhang, C. (2011) Neural network-based sliding mode adaptive control for robot manipulators. *Neurocomputing*. 74(14-15), 2377-2384. doi: 10.1016/j.neucom.2011.03.015.

Tilki, U. & Erüst, A. C. (2021) Robust Adaptive Backstepping Global Fast Dynamic Terminal Sliding Mode Controller Design for Quadrotors. *Journal of Intelligent & Robotic Systems*. 103(2), 1-12. doi: 10.1007/s10846-021-01475-2.

Tran, M. D. & Kang, H. J. (2017) Adaptive terminal sliding mode control of uncertain robotic manipulators based on local approximation of a dynamic system. *Neurocomputing*. 228, 231-240. doi: 10.1016/j.neucom.2016.09.089.

Truong, T. N., Vo, A. T. & Kang, H. J. (2021) A backstepping global fast terminal sliding mode control for trajectory tracking control of industrial robotic manipulators. *IEEE Access*. 9, 31921-31931. doi: 10.1109/ACCESS.2021.3060115.

Vijay, M. & Jena, D. (2018) Backstepping terminal sliding mode control of robot manipulator using radial basis functional neural networks. *Computers & Electrical Engineering*. 67, 690-707. doi: 10.1016/j.compeleceng.2017.11.007.

Wu, Q., Wang, X., Chen, B. & Wu, H. (2019) Neural network–based sliding-mode control of a tendon sheath–actuated compliant rescue manipulator. In *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*. 233(8), 1055-1066. doi: 10.1177/0959651819825984.

Yen, V. T., Nan, W. Y. & Cuong, P. Van (2019) Robust adaptive sliding mode neural networks control for industrial robot manipulators. *International Journal of Control, Automation and Systems*. 17(3), 783-792. doi: 10.1007/s12555-018-0210-y.

Yi, S. & Zhai, J. (2019) Adaptive second-order fast nonsingular terminal sliding mode control for robotic manipulators. *ISA Transactions*. 90, 41-51. doi: 10.1016/j.isatra.2018.12.046.

Zhai, J. & Xu, G. (2020) A novel non-singular terminal sliding mode trajectory tracking control for robotic manipulators. *IEEE Transactions on Circuits and Systems II: Express Briefs*. 68(1), 391-395. doi: 10.1109/TCSII.2020.2999937.

Zhang, J., Zhuang, J. & Du, H. (2009) Self-organizing genetic algorithm based tuning of PID controllers. *Information Sciences*. 179(7), 1007-1018. doi: 10.1016/j.ins.2008.11.038.

Zhang, L., Liu, L., Wang, Z. & Xia, Y. (2018) Continuous finite-time control for uncertain robot manipulators with integral sliding mode. *IET Control Theory & Applications*. 12(11), 1621-1627. doi: 10.1049/iet-cta.2017.1361.

Zhu, Q. (2021) Complete model-free sliding mode control (CMFSMC). *Scientific Reports*. 11(1), 22565. doi: 10.1038/s41598-021-01871-6.