

# Model based adaptive controller with grasshopper optimization algorithm for upper-limb rehabilitation robot

Aliaa Adnan Flaih Hassan, Ekhlas Hameed Karam, Muaayed F. Al-Rawi

Department of Computer Engineering, College of Engineering, Mustansiriyah University, Baghdad, Iraq

## Article Info

### Article history:

Received Sep 30, 2021

Revised Feb 10, 2022

Accepted Mar 11, 2022

### Keywords:

Grasshopper optimization algorithm

Lyapunov method

Model based adaptive controller

Rehabilitation robot

Two-link upper limb

## ABSTRACT

Model based adaptive controllers (MBACs) are considered one of the most common adaptive controllers that are used with robotic systems due to their ensuring nonlinear robust scheme with global asymptotic stability for controlling nonlinear systems. However, this controller requires precise mathematical models of the controlled systems. In this paper, an optimal model-based adaptive controller (OMBAC) is suggested for controlling a two-link upper limb rehabilitation robot. This controller, in the presence of model uncertainties, can guarantee the robustness of the rehabilitation robot. Although the OMBAC is an adaptive and model-based controller, some of its parameters need to be determined precisely. In this paper, these parameters are determined by the grasshopper optimization algorithm (GOA). The Lyapunov method is used to analyze the stability assurance of controlled rehabilitation. The results of the simulation for two tested trajectories (linear and nonlinear trajectories) demonstrate the efficiency of the suggested OMBAC with fast settling time, minimum error steady state, and very small overshoot.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Muaayed F. Al-Rawi

Department of Computer Engineering, College of Engineering, Mustansiriyah University

Palestine Street, PO Box: 14022, Baghdad, Iraq

Email: [muaayed@uomustansiriyah.edu.iq](mailto:muaayed@uomustansiriyah.edu.iq)

## 1. INTRODUCTION

Since the end of the 1990s, there has been a rush in the research and development of robotic devices for rehabilitation, especially for the neurological rehabilitation of post-stroke patients. Stroke is already the second leading cause of death in the world and the leading cause of acquired disability in adults [1]. Stroke survivors are typically disabled and have neurological deficits, and more than half will not fully recover basic motor skills for daily tasks, limiting their independence [2]–[5]. The ideal situation is for therapists to be physically active in patients' rehabilitation. However, there are a limited number of therapists [6], and many patients must perform rehabilitation exercises on their own. In these cases, it is not possible to monitor how well the patient is performing the treatment. The measured patient-specific variables are those that are important for tracking progress and improving upper and lower extremity movement control and coordination [7]–[9]. With continued support from the research, industrial and medical communities, new devices and applications will be designed and the use of these devices for diagnosis, treatment, and evaluation will be accepted throughout the recovery period. The main control techniques for controlling robotic devices are linear control and non-linear control. Linear controls are easy to implement but cannot handle non-linear disturbances of the system because rehabilitation robots are non-linear in nature, and non-linear control methodologies are more suitable for controlling robotic manipulators. Stability and robustness are two vital yet challenging topics for control systems. Several nonlinear controllers have been proposed,

and the most important suggested methods are adaptive controllers for obtaining greater repeatability and accuracy in the automated system's performance. Adaptive control is a type of control unit that has the ability to adjust itself to any changes in parameters that occur in the control system. There are different types of adaptive controllers. Some of them require parameter estimation, while others have a complex algorithm. Many adaptive schemes have been proposed for controlling rehabilitation robots. Some of these schemes are: Proietti *et al.* [10] proposed an exoskeleton controller based on adaptive techniques that can actively modulate the stiffness of the robotic device in function of the subject's activity. Based on Lyapunov stability, Lv *et al.* [11] proposed a model reference adaptive impedance control (MRAC) for lower limb rehabilitation robots. Zahid *et al.* [12] suggest an MRAC technique to control the movement of a DC motor for a one-DOF rehabilitation robot to reduce the robot's positioning error and make it useful for a wide range of stroke patients. Riani *et al.* [13] proposed the active integrated limb sliding posture control (AITSMC) technique that can be applied to forcefully control the exoskeletons of the upper limb to perform passive movement rehabilitation. Ting and Aiguo [14] studied the joint control of the stereotaxic arm for the purpose of robot-assisted upper limb rehabilitation therapy exercises by following the required path and making the system stabilize in the rehabilitation. Zhang *et al.* [15] suggest an adaptive fuzzy control scheme to adapt the control input for lower-limb exoskeleton rehabilitation according to the online performance. Abbasimoshai *et al.* [16] designed an adaptive fuzzy sliding mode controller (AFSMC) to control hand rehabilitation robots.

This controller improves system robustness by dealing with unknown parameters and uncertainties. This research focuses on designing a model base adaptive controller (MBAC), this is control system suitable for the case of unknown exactly or the parameters of the system are changed [17]. Using a grasshopper optimization algorithm (GOA) to tune MBAC control parameters for tracking the trajectory of a two-link upper limb rehabilitation robot based on a dynamic mathematical equation for a human two-joint upper-limb mode. GOA is used to tune the parameters of the suggested controller. Based on robot dynamic equations, the stability analyses of the closed-loop controlled system of both joints are proved by Lyapunov stability. The rest of this paper is organized as follows: section 2 presents the dynamic mathematical model of the two-link upper-limb rehabilitation robot, section 3 presents the suggested controller, section 4 presents the simulation results, section 5 presents the conclusions, and section 6 presents the conclusions.

## 2. DYNAMIC MATHEMATICAL MODEL OF UPPER ROBOTIC EXOSKELETON

In order to successfully design a controller and orient (orient) the robot to the desired location, it is necessary to describe the mathematical model of the robot. As shown in Figure 1, via assuming smooth human-machine interaction, the upper-extremity dynamics of both the human limb and the robotic exoskeleton are treated as rigid links joined by joints, and the two-link model is limited in the sagittal plane [18], [19]. In this model, the frictional components operating on the exoskeleton and human joints, as well as other atypical dynamics are ignored.

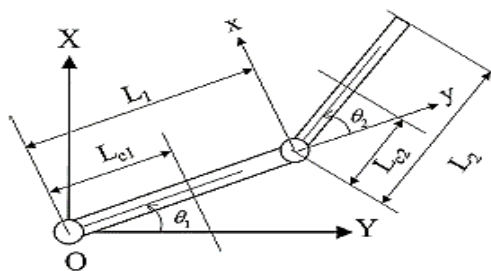


Figure 1. A two-link manipulator that appears like a human upper limb [18]

In Figure 1, the letters 1 and 2 indicate the parameters of the first (upper arm) and second (forearm) links, respectively. The letter L stands for the length of the limb as well as the exoskeleton. Lc is the angular position of the couplings as well as the length of the terminal parts and the exoskeleton about the centroid axis. The Euler-Lagrange formula is used to derive the equation of motion for the dynamic system of the upper limb. The nonlinear differential equations can be written as [20], expressed as (1).

$$\tau = D(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) + \tau_d \quad (1)$$

Where  $\tau$  is the operational torque vector,  $D$  is a  $2 \times 2$  inertial matrix of the extremities and exoskeleton,  $C$  is the Coriolis and the central torque vector,  $G$  is the gravitational torque vector while  $\tau_d$  is the external torque vector. The (1) it can be rewritten as (2):

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} - \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} - \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} - \begin{bmatrix} \tau_{d1} \\ \tau_{d2} \end{bmatrix} \right\} \quad (2)$$

where the element of the  $D$  inertial matrix is expressed as (3).

$$\begin{aligned} D_{11} &= m_1 L_{c1}^2 + I_1 + m_2 (L_{c2}^2 + L_1^2 + 2L_{c2} L_1 \cos \theta_2) + I_2, \\ D_{12} = D_{21} &= m_{12} L_1 L_{c2} \cos \theta_2 + m_2 L_{c2}^2 + I_2, \quad D_{22} = m_2 L_{c2}^2 + I_2 \end{aligned} \quad (3)$$

The element of the coriolis matrix are expressed as (4).

$$\begin{aligned} C_{11} &= -m_2 L_1 L_2 (2\dot{\theta}_2) \sin \theta_2, \quad C_{12} = -m_2 L_1 L_2 \dot{\theta}_2 \sin \theta_2, \\ C_{21} &= m_2 L_1 L_2 \dot{\theta}_1 \sin \theta_2, \quad C_{22} = 0 \end{aligned} \quad (4)$$

While the elements of gravitational term are expressed as (5):

$$\begin{aligned} G_1 &= (m_1 + m_2) g L_1 \cos \theta_1 + m_2 g L_2 \cos(\theta_1 + \theta_2), \\ G_2 &= m_2 g L_2 \cos(\theta_1 + \theta_2) \end{aligned} \quad (5)$$

where  $m$  is a sum of both masses, while  $I$  is the moment of mass from the exoskeleton's inertia and the extremities' inertia, and  $g$  is the gravitational constant taken as  $9.81 \text{ m/s}^2$ .

### 3. MODEL BASED ADAPTIVE CONTROLLER DESIGN

In order to design the suggested MBAC, the equation of dynamic system (1) must be rewritten to (6):

$$f_{pi}(\theta, u, t) = f_i(\theta, \dot{\theta}) + B_i(\theta, t)u_i(\theta, t) + \tau_{di}(t) \quad (6)$$

where  $i$  is robot link number (1, 2), and  $f(\theta, t)$  and  $B(\theta, t)u(\theta, t)$  are expressed as (7) and (8).

$$f(\theta, t) = D^{-1}(\theta) \{ C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \} \quad (7)$$

$$B(\theta, t)u(\theta, t) = D^{-1}(\theta) \tau(\theta, t) \quad (8)$$

It is required to design an adaptive controller based on the dynamic model of the rehabilitation robot in order to make each robot link follow the desired input trajectory. The procedure for designing this controller for each link ( $i = 1, 2$ ) is represented by the following steps:

Step 1: Define the equation of the desired model as (9):

$$\dot{q}_{di}(t) = A_i q_{di}(t) + B_i r_{di}(t) \quad (9)$$

where  $i=1,2$  is link number, the  $q_{di}$  is  $2 \times 1$  desired model state vector,  $A_i$  is  $2 \times 2$  constant matrix,  $B_i$  is  $2 \times 1$  constant matrix, and  $r_{di}$  is  $2 \times 1$  desired input trajectory. The suitable desired model is selected as (10):

$$\begin{bmatrix} \dot{q}_{d1}(t) \\ \dot{q}_{d2}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} q_{d1}(t) \\ q_{d2}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} \begin{bmatrix} r_{d1}(t) \\ r_{d2}(t) \end{bmatrix} \quad (10)$$

where  $\zeta$  is damping ratio and  $\omega_n$  is desired natural frequency. The value  $\omega_n$  and  $\zeta$  should be selected such that the eigen-values of ( $A_i$ ) are negative roots, and hence the equilibrium point of this model has an asymptotically stable.

Step 2: Define the difference between  $q_{di}$  and  $q_i$  as error signal ( $e_i(t)$ ) by (11).

$$e_i(t) = q_{di}(t) - q_i(t) \quad (11)$$

The derivative of the  $e_i(t)$  is given by (12).

$$\begin{aligned}\dot{e}_i(t) &= \dot{q}_{di}(t) - \dot{q}_i(t) \\ &= A_i q_{di}(t) + B_i r_{di}(t) - f_{mi}(\theta, u, t) \\ &= A_i e_i(t) + A_i q_i(t) - f_{mi}(\theta, u, t) + B_i r_{di}(t)\end{aligned}\quad (12)$$

Step 3: Select Lyapunov function for each link ( $i=1, 2$ ) as (13):

$$V_i(e_i) = e_i(t)^T P_i e_i(t) \quad (13)$$

where  $P_i$  is a real positive-definite symmetry matrix. When the derivative of  $V_i(e_i)$  is taken with respect time, we obtain as (14), where expressed as (15).

$$\begin{aligned}\dot{V}_i(e_i) &= \dot{e}_i^T P_i e_i + e_i^T P_i \dot{e}_i \\ &= [e_i^T A_i^T + q_i^T A_i^T - f_{mi}^T(\theta, u, t) + r_{di}^T B_i^T] P_i e_i + e_i^T P_i [A_i e_i + A_i q_i - f_{mi}(\theta, u, t) + B_i r_{di}] \\ &= e_i^T (A_i^T P_i + P_i A_i) e_i + 2N_i\end{aligned}\quad (14)$$

$$N_i = e_i^T P_i [A_i q_i - f_{mi}(\theta, u, t) + B_i r_{di}] \quad (15)$$

Since  $M_i = -(A_i^T P_i + P_i A_i)$ , therefore (14) can be written as (16).

$$\dot{V}_i(e_i) = -e_i^T Q_i e_i + 2N_i \quad (16)$$

According to theorem 2 of [21], the controlled system is globally asymptotically stable (GAS) in the sense of Lyapunov if the  $V_i(e_i)$  is positive definite due to the positive selected  $P_i$  matrix and  $\dot{V}(x) < 0 \forall x \neq 0$  and  $\dot{V}(0) = 0$  when the matrix  $(-M_i)$  is a negative definite. The control signal  $u_i(\theta, t)$  should be chosen such that the scalar  $N_i$  is negative [21]. The (16) is to satisfy if matrix  $M_i$  is selected as diagonal matrix.

$M_i = \text{positive definite} = \begin{bmatrix} m_{11i} & 0 \\ 0 & m_{22i} \end{bmatrix}$  and hence (16) became (17).

$$\dot{V}_i(e_i) = -(m_{11i} e_{1i}^2 + m_{22i} e_{2i}^2) + 2N_i \quad (17)$$

The scalar  $N_i$  according to the desired model (9) and the robot model (6) will be (18).

$$\begin{aligned}N_i &= [e_{1i} \quad e_{2i}] \begin{bmatrix} p_{11i} & p_{12i} \\ p_{21i} & p_{22i} \end{bmatrix} \\ &\quad \left( \begin{bmatrix} 0 & 1 \\ -\omega_{ni}^2 & -2\zeta_i \omega_{ni} \end{bmatrix} \begin{bmatrix} q_{1i} \\ q_{2i} \end{bmatrix} - \begin{bmatrix} 0 \\ f_i(\theta, \dot{\theta}) \end{bmatrix} - \begin{bmatrix} 0 \\ B_i(\theta, t) u_i(\theta, t) \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_{ni}^2 r_{di} \end{bmatrix} \right) \\ N_i &= (e_{1i} p_{12i} + e_{2i} p_{22i}) [-\omega_{ni}^2 q_{1i} - 2\zeta_i \omega_{ni} q_{2i} + \omega_{ni}^2 r_{di} - B_i(\theta, t) u_i(\theta, t) - f_i(\theta, \dot{\theta})]\end{aligned}\quad (18)$$

According to the (18), the control law for each robot link  $u_i(\theta, t)$  can be designed as (19):

$$u_i(\theta, t) = [-\omega_n^2 q_{1i} - 2\zeta \omega_n q_{2i} + (k_i \tanh(e_{1i} p_{12i} + e_{2i} p_{22i})) + \omega_{ni}^2 r_{di} - f_i(\theta, \dot{\theta})] / B_i(\theta, t) \quad (19)$$

where  $k_i$  is optimal constant value and  $\tanh(\cdot)$  is hyperbolic tangent function. According to (19) the equation of  $N_i$  is becomes (20).

$$N_i = e_{1i} p_{12i} + e_{2i} p_{22i} [f_i(\theta, \dot{\theta}) - k_i \tanh(e_{1i} p_{12i} + e_{2i} p_{22i})] < 0 \quad (20)$$

In this paper, in order to make the controller  $u_i(\theta, t)$  became more efficient, the state  $q_{1i}$  and  $q_{2i}$  in (19) is added with the error signal  $(e_{1i}, e_{2i})$ , therefore this equation becomes (21).

$$u_i(\theta, t) = \left[ \begin{aligned} & -\omega_n^2 (q_{1i} + e_{1i}) - 2\zeta \omega_n (q_{2i} + e_{2i}) \\ & + (k_i \tanh(e_{1i} p_{12i} + e_{2i} p_{22i})) + \omega_{ni}^2 r_{di} - f_i(\theta, \dot{\theta}) \end{aligned} \right] / B_i(\theta, t) \quad (21)$$

The control parameters  $(K, k_1, p_{11}, p_{12}, \omega_{n1})$  for link 1 and parameters  $(K, k_2, p_{21}, p_{22}, \omega_{n2})$  of link 2 are obtained by GOA which is explained in following section.

**4. GRASSHOPPER OPTIMIZATION ALGORITHM**

Saremi *et al.* [22] introduced the GOA algorithm, which is a new and fascinating swarm intelligence algorithm that simulates grasshopper swarming behaviour. Grasshoppers are insects that cause havoc on crop production and agriculture [22], [23]. Their life cycle is divided into two stages: nymph and adulthood. Small steps and moderate movements describe the nymph phase, whereas long-range and rapid movements represent the maturity phase [24]. The intensification and divarication phases of GOA are defined by nymph and adult motions. The mathematical model for grasshopper swarming behaviour is as [22], [23], expressed as (22).

$$X_i = S_i + W_i + Z_i \tag{22}$$

In the (22),  $X_i$  represents the grasshopper's  $i^{th}$  position,  $S_i$  represents social participation,  $W_i$  represents the force of gravity on a grasshopper, and  $Z_i$  represents air-advection. In the preceding equation, randomness is produced by (23).

$$X_i = h_1 S_i + h_2 W_i + h_3 Z_i \tag{23}$$

Where  $h_1, h_2$  and  $h_3$  are a sequence of random numbers range of 0 to 1.  $S_i$  is designed as (24).

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^N s(d_{ji}) \widehat{d}_{ji} \tag{24}$$

The distance between  $i^{th}$  and  $j^{th}$  grasshopper is computed using (25), In addition as (26).

$$d_{ji} = |x_j - x_i| \tag{25}$$

$$\widehat{d}_{ji} = \frac{x_j - x_i}{d_{ji}} \tag{26}$$

In addition, the strength of the social force  $s$  is described as (27):

$$s(r) = F y^{\frac{-d}{l}} - y^{-d} \tag{27}$$

where  $F$  denotes the attractive force,  $d$  denotes the distance, and  $l$  denotes the attraction measure. The (22) represent  $W$  component is written as (28):

$$W_i = -b \hat{y}_b \tag{28}$$

where  $b$  is the gravitational force, the negative sign shows its orientation toward the center of the Earth, while  $\hat{y}_b$  is the unit vector toward the Earth. Now, the  $Z$  component in (22) is given as (29):

$$Z_i = c \hat{y}_w \tag{29}$$

where  $c$  is the continuous wind drift and  $\hat{y}_w$  shows the wind direction unit vector. Putting the values of  $S, W$  and  $Z$  in (22), we get (30).

$$X_i = \sum_{\substack{j=1 \\ j \neq i}}^N s(|x_j - x_i|) \frac{x_j - x_i}{d_{ji}} - b \hat{y}_b + c \hat{y}_w \tag{30}$$

We use the (30) to describe the interaction of grasshoppers in a swarm in free space and in simulation. Table 1 shows the steps involved in the GOA algorithm. The cost function used in this research is integral time-weighted squared error (ITSE), which is described by (31):

$$Fitness = ITSE = \int_0^t t * Error(t)^2 \tag{31}$$

where  $t$  is the error between the desired input and the response in each link.

Table 1. General GOA pseudo code [25]

Step	Description
1	The objective function $F = \text{MAX}_{pp}$ Create an initial population of grasshoppers $x_i$ with the values $1, 2, \dots, n$ Calculate each grasshopper's fitness $R =$ the most effective search agent
2	While the stopping criteria have not been met do
3	Update $c_1$ and $c_2$ For each grasshopper in the population do Normalize the distances between grasshoppers in [1, 4] Update the position of the grasshopper If required, update bounds of grasshopper End for
4	If there is a better solution, update $R$ End while
5	Output the $R$

## 5. SIMULATION RESULTS

To demonstrate the efficiency of the suggested controller with MATLAB software version R2020a v facility, two simulation scenarios for the upper limb rehabilitation robot is run with 10% parameter uncertainty in the function parameters for both linear (step) and nonlinear trajectories and  $0.01 \sin(20t)$  disturbance. The parameters of upper limb rehabilitation are given in Table 2. The GOA algorithm parameters that are considered here are listed in Table 3. The optimal controller parameters, which are obtained by the GOA, are given in Table 4.

Table 2. Upper-limb robot parameters [24]

Upper-limb parameters	Parameter	Actual value	Units
Limb and exoskeleton masses	$m_1$	2.25	kg
	$m_2$	1.47	kg
Limb lengths	$L_1$	0.34	m
	$L_2$	0.25	m
Center of mass	$L_{c1}$	0.25	m
	$L_{c2}$	0.125	m
Mass moment of inertia for exoskeleton and limbs	$I_1$	0.2505	$\text{kg m}^2$
	$I_2$	0.0925	$\text{kg m}^2$

Table 3. The GOA parameters

GOA parameters	Value
Number of search agents	25
Max iteration	5
Intensity of attraction (f)	0.5
Attractive length scale (l)	1.5

Table 4. The optimal parameters for the suggested controller obtained by GOA

Controller parameters of two links	Lower bound of value	Upper bound of value	Optimal value of link1	Optimal value of link2
K parameter with saturation	30	60	56.1099	49.8942
$p_{11}$	5	15	13.7468	11.8405
$p_{12}$	1	3	1.0307	1.0260
$\omega_n$	90	115	54.5620	55.3889
k parameter of feedback	8	20	12.8926	14.2946

### 5.1. Linear path with 10% uncertainties and nonlinear disturbance

The response of an upper limb rehabilitation robot simulation by the proposed controller with uncertainty and disturbance is shown in Figures 2-5 as the position of the shoulder link, the elbow link for linear path, the control signal of link 1 for linear path, and the control signal of link 2 for linear path, respectively. These results indicate that the robot with the suggested controller performs better, with the robot following the intended path extremely quickly. With a smooth control signal, there is no steady-state error. The simulation results and assessment parameters for the suggested controller are shown in Table 5.

Table 5. Parameters of evaluating the simulation result of the proposed controller

Parameters	Link1 (shoulder)	Link2 (elbow)
$M_p$ (%)	0	0
$t_s$	0.1888	0.1976
$es.s$	0.0023	-0.0005
$t_r(sec.)$	0.1085	0.1183

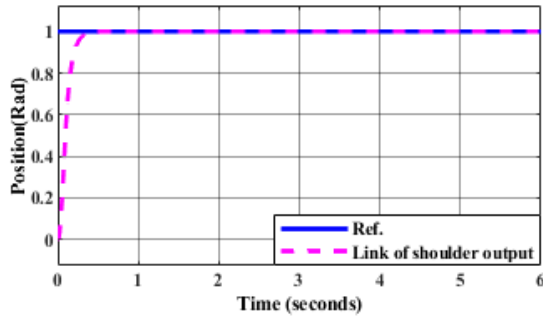


Figure 2. The position of shoulder link

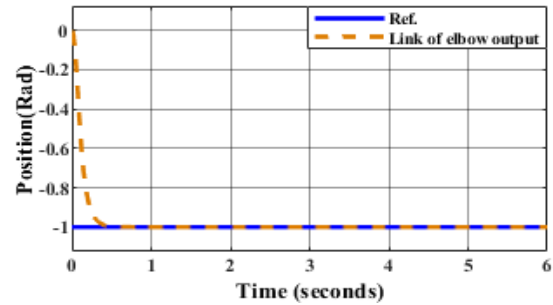


Figure 3. The elbow link for linear path

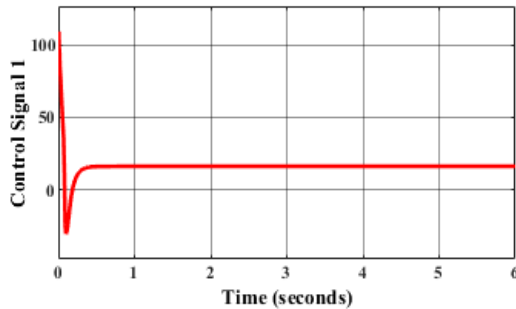


Figure 4. The control signal of link 2 for linear path

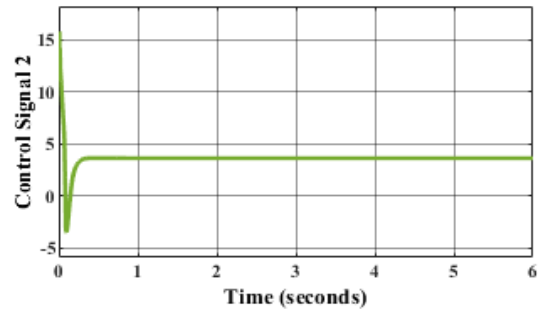


Figure 5. The control signal of link 2 for linear path

**5.2. Nonlinear path with 10% uncertainties and nonlinear disturbance**

A sinusoidal input is introduced into the system at both joints to evaluate the tracking performance of the suggested control approach. The insertions at the elbow (joint 1) and shoulder (joint 2) joints are then repeated. In order to investigate the robustness of both control architectures, one of type disturbance  $0.01 \sin(20t)$  was placed in the system with an uncertainty of 10% in the coefficients of the function to investigate the robustness of both control topologies. As a result of the simulation study, the position of the shoulder link, the elbow link for nonlinear path, the control signal of link 1 for nonlinear path, and the control signal of link 2 for nonlinear path are indicated in Figures 6-9 respectively.

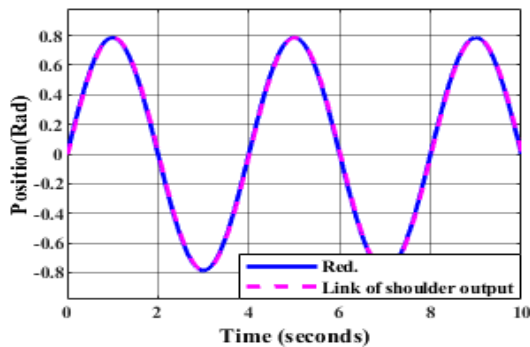


Figure 6. The position of shoulder link

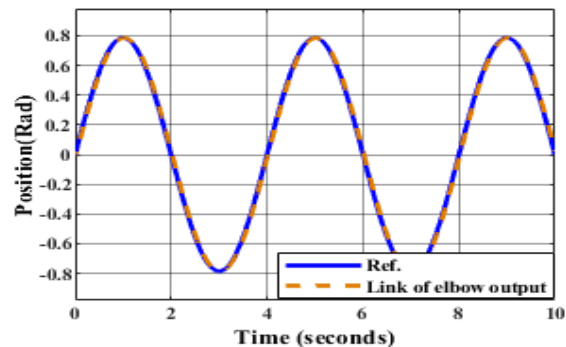


Figure 7. The elbow link for nonlinear path

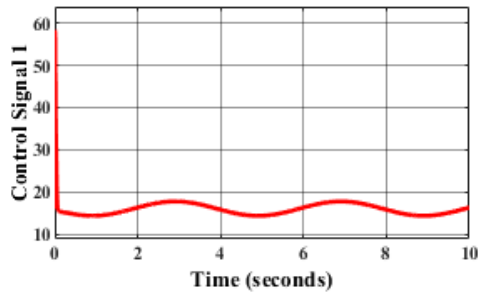


Figure 8. The control signal of link 1 for nonlinear

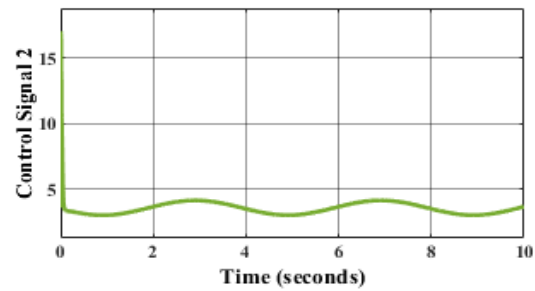


Figure 9. The control signal of link 2 for nonlinear path

## 6. CONCLUSION

The main purpose of this research was to create a model base adaptive controller (MBAC) that could follow desired paths and make the two-link upper limb rehabilitation robot work better. The parameters of this controller are obtained by the grasshopper optimization algorithm. The efficiency of the suggested controller is indicated by the measured results' parameters, where the actual response of both links followed the desired linear path with zero steady state error, and with no phase shift with the desired nonlinear path despite the existence of 10% parameter uncertainty in  $f_i(\theta, \dot{\theta})$  function and external disturbance  $(0.01 \sin 20t)$  in the linear case in the in achieving in the controlled case. These results indicate the effectiveness of the proposed controller and suggest that their capabilities be studied in more complex situations and physical implementations.

## REFERENCES




- [1] T. Proietti, V. Crocher, A. Roby-Brami, and N. Jarrasse, "Upper-limb robotic exoskeletons for neurorehabilitation: a review on control strategies," *IEEE Reviews in Biomedical Engineering*, vol. 9, pp. 4–14, 2016, doi: 10.1109/RBME.2016.2552201.
- [2] G. E. Gresham, T. E. Fitzpatrick, P. A. Wolf, P. M. McNamara, W. B. Kannel, and T. R. Dawber, "Residual disability in survivors of stroke — The Framingham study," *New England Journal of Medicine*, vol. 293, no. 19, pp. 954–956, Nov. 1975, doi: 10.1056/nejm197511062931903.
- [3] A. K. Welmer, L. W. Holmqvist, and D. K. Sommerfeld, "Limited fine hand use after stroke and its association with other disabilities," *Journal of Rehabilitation Medicine*, vol. 40, no. 8, pp. 603–608, 2008, doi: 10.2340/16501977-0218.
- [4] S. M. Lai, S. Studenski, P. W. Duncan, and S. Perera, "Persisting consequences of stroke measured by the stroke impact scale," *Stroke*, vol. 33, no. 7, pp. 1840–1844, Jul. 2002, doi: 10.1161/01.STR.0000019289.15440.F2.
- [5] L. W. Jian and S. N. W. Shamsuddin, "The design of virtual lower limb rehabilitation for post-stroke patients," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 1, pp. 544–552, Oct. 2019, doi: 10.11591/ijeecs.v16.i1.pp544-552.
- [6] Heart and Stroke Foundation, "2014 report on the health of Canadians," 2014. [Online]. Available: <https://www.heartandstroke.ca/-/media/pdf-files/canada/2017-heart-month/heartandstroke-reportonhealth-2014.aspx?la=en&hash=9860137823BF864C3DE8B4CBB9F57826A7C40C3>.
- [7] A. Roy *et al.*, "Measurement of human ankle stiffness using the anklebot," in *2007 IEEE 10th International Conference on Rehabilitation Robotics, ICORR'07*, Jun. 2007, pp. 356–363, doi: 10.1109/ICORR.2007.4428450.
- [8] P. S. Lum, C. G. Burgar, P. C. Shor, M. Majmundar, and M. Van der Loos, "Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke," *Archives of Physical Medicine and Rehabilitation*, vol. 83, no. 7, pp. 952–959, Jul. 2002, doi: 10.1053/apmr.2001.33101.
- [9] K. P. Westlake and C. Patten, "Pilot study of Lokomat versus manual-assisted treadmill training for locomotor recovery post-stroke," *Journal of NeuroEngineering and Rehabilitation*, vol. 6, no. 1, p. 18, Dec. 2009, doi: 10.1186/1743-0003-6-18.
- [10] T. Proietti, N. Jarrasse, A. Roby-Brami, and G. Morel, "Adaptive control of a robotic exoskeleton for neurorehabilitation," in *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, Apr. 2015, pp. 803–806, doi: 10.1109/NER.2015.7146745.
- [11] X. Lv, J. Han, C. Yang, and D. Cong, "Model reference adaptive impedance control in lower limbs rehabilitation robot," in *2017 IEEE International Conference on Information and Automation, ICIA 2017*, Jul. 2017, pp. 254–259, doi: 10.1109/ICInfA.2017.8078915.
- [12] J. Zahid, K. X. Khor, C. F. Yeong, E. L. M. Su, and F. Duan, "Adaptive control of DC motor for one-DOF rehabilitation robot," *Journal of Electrical Engineering*, vol. 16, no. 3, pp. 1–5, 2017, doi: 10.11113/elektrika.v16n3.29.
- [13] A. Riani, T. Madani, A. Benallegue, and K. Djouani, "Adaptive integral terminal sliding mode control for upper-limb rehabilitation exoskeleton," *Control Engineering Practice*, vol. 75, pp. 108–117, Jun. 2018, doi: 10.1016/j.conengprac.2018.02.013.
- [14] W. Ting and S. Aiguo, "An adaptive iterative learning based impedance control for robot-aided upper-limb passive rehabilitation," *Frontiers Robotics AI*, vol. 6, no. Jun, pp. 1–11, Jun. 2019, doi: 10.3389/frobt.2019.00041.
- [15] X. Zhang *et al.*, "Novel design and adaptive fuzzy control of a lower-limb elderly rehabilitation," *Electronics (Switzerland)*, vol. 9, no. 2, p. 343, Feb. 2020, doi: 10.3390/electronics9020343.
- [16] A. Abbasimoshaei, M. Mohammadimoghaddam, and T. A. Kern, "Adaptive fuzzy sliding mode controller design for a new hand rehabilitation robot," in *Haptics: Science, Technology, Applications*, I. Nisky, J. Hartcher-O'Brien, M. Wiertelowski, and J. Smeets, Eds. 2020, pp. 506–517.






- [17] M. C. Nguyen, D. P. Vuong, and T. T. Nguyen, "The MRAC based-adaptive control system for controlling the speed of direct current motor," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 2, pp. 723–728, Aug. 2020, doi: 10.11591/ijeecs.v19.i2.pp723-728.
- [18] Z. Taha, A. P. P. A. Majeed, M. Y. W. P. Tze, M. A. Hashem, I. M. Khairuddin, and M. A. M. Razman, "Modelling and control of an upper extremity exoskeleton for rehabilitation," *IOP Conference Series: Materials Science and Engineering*, vol. 114, no. 1, Feb. 2016, p. 012134, doi: 10.1088/1757-899X/114/1/012134.
- [19] S. Aole, I. Elamvazuthi, L. Waghmare, B. Patre, and F. Meriaudeau, "Non-linear active disturbance rejection control for upper limb rehabilitation exoskeleton," *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 235, no. 5, May 2021, pp. 606–632, doi: 10.1177/0959651820954575.
- [20] J. J. Craig, *Introduction to robotics: mechanics and control*, 3rd ed. New Jersey: Pearson Education International, 2005.
- [21] K. Ogata, *Modern control engineering*, 3rd ed. New Jersey: Prentice-Hall, Inc., 1996.
- [22] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, Mar. 2017, doi: 10.1016/j.advengsoft.2017.01.004.
- [23] A. A. Ewees, M. A. Elaziz, Z. Alameer, H. Ye, and Z. Jianhua, "Improving multilayer perceptron neural network using chaotic grasshopper optimization algorithm to forecast iron ore price volatility," *Resources Policy*, vol. 65, p. 101555, Mar. 2020, doi: 10.1016/j.resourpol.2019.101555.
- [24] H. E. J. Veeger, B. Yu, K. N. An, and R. H. Rozendal, "Parameters for modeling the upper extremity," *Journal of Biomechanics*, vol. 30, no. 6, pp. 647–652, Jun. 1997, doi: 10.1016/S0021-9290(97)00011-0.
- [25] B. V. K. Ram and N. Chidambararaj, "Grasshopper optimization algorithm utilized Xilinx controller for maximum power generation in photovoltaic system," *Evolving Systems*, vol. 12, no. 4, pp. 885–898, Dec. 2021, doi: 10.1007/s12530-020-09333-6.

## BIOGRAPHIES OF AUTHORS






**Aliaa Adnan Flaih Hassan**    completed her B.Sc. and M.Sc. degrees in computer engineering at Mustansiriyah University, Baghdad, Iraq. Her research interests are in machine learning, artificial intelligence systems, image processing, and robotics. She can be contacted at email: aliaaadnan96@gmail.com.



**Ekhlas Hameed Karam**    completed her B.Sc., M.Sc., and Ph.D. degrees in computer engineering at the University of Technology, Baghdad, Iraq. Her research interests are in robotic systems, different controller designs, optimization methods, image processing, and FPGA. She can be contacted at email: ek\_karam@uomustansiriyah.edu.iq.



**Muaayyed F. Al-Rawi**    completed his B.Sc., M.Sc., and Ph.D. degrees in communications & electronics at Baghdad University, Baghdad, Iraq. His research interests are in wireless communications systems, DSP, and image processing. He can be contacted at email: muaayyed@uomustansiriyah.edu.iq.