Design of Modified Adaptive PID Controller for Lower Limb Rehabilitation Robot based on Grey Wolf Optimization Algorithm

Noor Sabah*
Electrical Engineering Department, Faculty of Engineering, University of Wasit, Iraq.
E-mail: noors@uowasit.edu.iq

Ekhlas Hameed
Computer Engineering Department, Faculty of Engineering, Mustansiriyah University, Iraq.

Muayed S AL-Huseiny
Electrical Engineering Department, Faculty of Engineering, University of Wasit, Iraq.

Received August 07, 2021; Accepted November 24, 2021
ISSN: 1735-188X
DOI: 10.14704/WEB/V19I1/WEB19023

Abstract

The traditional Proportional-Integral-Derivative (PID) controller is a feedback control loop that is commonly used in industrial control systems with fixed parameters, whereas the adaptive PID (APID) controller is based on the analysis of traditional PID controllers. It utilizes an online parameter adjustment method built on the state of the system resulting in better system adaptability. In this paper, the APID controller that is suggested by (Ebel, 2011) is used firstly to control a 2-degree of freedom (DOF) lower limb rehabilitation robot. The structure of this controller is then modified to perform a Modified Adaptive PID (MAPID) control in order to improve the efficiency of APID controller and hence improve the performance of the rehabilitation robot. The parameters of APID and the suggested MAPID controllers are optimized by using Grey Wolf Optimization (GWO) algorithm. Linear and non-linear desired trajectories are used to test the performance of the controlled rehabilitation robot. Simulation results show that the obtained performance of the rehabilitation robot is more efficient with the MAPID than with the APID having no overshoot and very small steady state error. The controller has settling time of (0.463) and (0.851) seconds, and rise time of (0.485) and (0.752) seconds respectively for link1 and link2.

Keywords

Modified Adaptive PID Controller, Grey Wolf Optimization Algorithm, Lower limb, Rehabilitation Robot.

Introduction

Stroke is a medical condition that occurs due to the obstruction of blood flow in the brain. This results in a series of devastating biochemical reactions which can lead to the death of brain neurocytes. When a stroke hits, it may put the patient in a state of post-stroke paralysis. This is manifested by reduced memory and speech capabilities, and further leading to impaired motor ability of the lower limps. Stroke rehabilitation approaches dedicate their efforts to enabling the patients restore some or all of their physical capabilities effectively and independently (Saryanto, Cahyadi and Herianto, 2014). Same efforts apply to ease the lives of traumatic and spinal cord injuries' patients. Such rehabilitation programs would normally rely on the skills of a trained physical therapist. With the increase of modern life's dangers and consequences as well as the global shortage of trained personnel, the mechanization/ automation of physical rehabilitation procedures would significantly improve such programs while preserving patients' private lives (Ju *et al.*, 2005). A recent trend in this direction is the development in robotics design robot mediated./aided therapy. This approach can be particularly useful when traditional physical therapy backfires (Veneman *et al.*, 2007). Another important aspect of this technology in the era of aging population is the assistance of the increasing elderly people to continue their lives independently (Bien *et al.*, 2004). The implementation of a stable controller within the robot-assisted rehabilitation system is a crucial requirement for patients' safety and physical functionality. As such, nonlinearities and uncertainties in the robot dynamics while attached to the patient's extremities must be in order to achieve the desirable stale results (Torabi, Sharifi and Vossoughi, 2017). The proportional-integral-derivative (PID) control strategy is implemented in a wide range of control engineering solutions. It is preferred for being simple and effectiveness while being simple. Yet, fine tuning of this controller by selection of suboptimal values for its parameters, the proportional (K_p), integral (K_i), and derivative (K_d) gains is deemed an area active of research. In conventional settings the gains of PID controllers are fixed to predetermined values. However, adaptive control law can be implemented to adjust these parameters continuously (Rahrooh, Motlagh and Buchanan, 2005). Both conventional PID and adaptive PID controllers are utilized effectively rehabilitation robot procedures. (Wu *et al.*, 2012) proposed RLSESN-based adaptive PID control for a novel wearable rehabilitation robotic hand driven by PM-TS actuators for improving the trajectory tracking performance of the rehabilitation robotic hand. (Gilbert, Zhang and Yin, 2016) proposed a modelling and design of PID control system for lower limb rehabilitation exoskeleton robot that decreases tracking errors to achieve desired trajectory. Their published results show a good trajectory tracking performance for exoskeleton in accordance with the dynamics of human motion. (Zahid *et al.*, 2017)

proposed reference model adaptive PID controller for 1-DOF rehabilitation robot to reduce positioning error and make the robot beneficial for a wide range of stroke patients.

This paper focuses on designing a Modified Adaptive Proportional-Integral-Derivative (MAPID) Controller. The proposed controller is inspired by the APID controller presented in (Ebel, 2011). The parameters of the APID and the suggested MAPID controllers are separately optimized by Grey Wolf Optimization (GWO) algorithm for tracking the trajectory of a two-link lower-limb rehabilitation robot. The system employs a dynamic equation for a human two-joint during-walk lower-limb model. Stability analyses of both joints of a closed-loop controlled system based on the dynamic robot equations are investigated by Lyapunov stability.

The rest of this paper is organized as follows: the dynamic mathematical model of the two-link lower-limb rehabilitation robot is provided in section 2, the implemented controller is detailed in section 3, the optimization algorithm is presented in section 4, simulation results are discussed in section 5, and, finally, the paper concludes in section 6.

Dynamic Model of Lower Limb Rehabilitation Robot

The structure of a two degree of freedom (2-DOF) rehabilitation robot is shown in Fig. 1, this robot consists of two links with two joints of the lower limb: a joint at the hip for link1 and a joint at the knee for link2. The dynamic model of this robot was derived by (Rezage and Tokhi, 2016) based on anthropometric data (described by Winter (2009)) for a subject with 74 kg in weight and 1.69 m in height [(Alshatti, 2019), (Winter, 2009)].

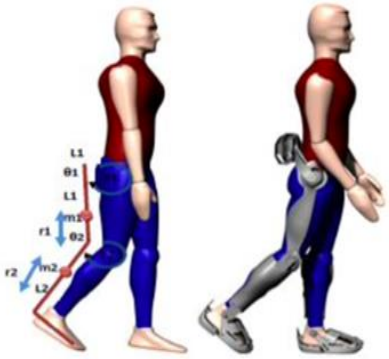


Figure 1 2-DOF Rehabilitation Robot (Rezage and Tokhi, 2016)

Mathematically, the dynamics of such a robotic system can be approximated by the following form in Eq. (1): (Rezage and Tokhi, 2016).

$$M(\theta)\ddot{\theta} + C\left(\theta, \dot{\theta}\right)\dot{\theta} + G(\theta) = u(t) \qquad (1)$$

where θ,$\dot{\theta}$ and $\ddot{\theta}$ are the vectors of the joint's angle, angular velocity and acceleration, respectively. $M(\theta) \in R^{2*2}$ is the inertia matrix of the lower limb, $C(\theta, \dot{\theta}) \in R^{2*2}$ is the vector of centrifugal and Coriolis torque of human limb, $G(\theta) \in R^{2*1}$ is the vector of gravitational torque of the human limb, $u(t)$ indicates the control signal.

The robot dynamics, are represented in Eq. (2):

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} u(t)_1 \\ u(t)_2 \end{bmatrix} \quad (2)$$

The elements of the inertia matrix $M(\theta)$ are presented in Eq. (3):

$$\begin{gathered} M_{11} = I_1 + I_2 + m_1(L_{c1})^2 + m_2(L_1)^2 + m_2(L_{C2})^2 + 2m_2L_1L_{C2}\, cos(\theta_2) \\ M_{12} = M_{21} = I_2 + m_2(L_{C2})^2 + \ m_2L_1L_{C2}\, cos(\theta_2) \\ M_{22} = I_2 + m_2(L_{C2})^2 \end{gathered} \quad (3)$$

The elements of $C(\theta, \dot{\theta})$ are given by Eq. (4):

$$\begin{gathered} C_{11} = -m_2L_1L_{c2}\, sin(\theta_2)\dot{\theta}_2 \\ C_{12} = -m_2L_1L_{c2}\, sin(\theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \\ C_{21} = m_2L_1L_{c2}\, sin(\theta_2)\dot{\theta}_1 \\ C_{22} = 0 \end{gathered} \quad (4)$$

The parameter of the gravitational vector $G(\theta)$ are given by Eq. (5):

$$\begin{gathered} G_1 = m_1L_{c1}g\, sin(\theta_1) + m_2gL_1\, sin(\theta_1) + \ m_2gL_{c2}\, sin(\theta_1 + \theta_2) \\ G_2 = m_2gL_{c2}\, cos(\theta_1 + \theta_2) \end{gathered} \quad (5)$$

The variables of these equations and physical parameters are defined in Table 1.

Table 1 The variables and physical parameters for lower limb rehabilitation robot

Parameter	Definition	Unit	Value
L_1	Length of link 1	m	0.54
L_2	Length of link 2	m	0.48
L_{c1}	Link (1) center of mass	m	0.2338
L_{c2}	Link (2) center of mass	m	0.241
m_1	Mass of link 1	Kg	8
m_2	Mass of link 2	Kg	3.72
I_1	Inertia of link 1	$Kg.m^2$	0.42
I_2	Inertia of link 2	$Kg.m^2$	0.07
g	Gravity acceleration	m/s^2	9.8
θ_1	Link (1) angular displacement	Rad	/
θ_2	Link (2) angular displacement	Rad	/
$\dot{\theta}_1$	Link (1) angular velocity	Rad/s	/
$\dot{\theta}_2$	Link (2) angular velocity	Rad/s	/
$\ddot{\theta}$	Angular acceleration	Rad/s^2	/

Adaptive PID (APID) Controller Design

The structure of the APID controller that is suggested by (Ebel, 2011) is used here to build an adaptive controller for the two link lower limb rehabilitation robot. The block diagram for the designed controller is shown in Fig. (2).

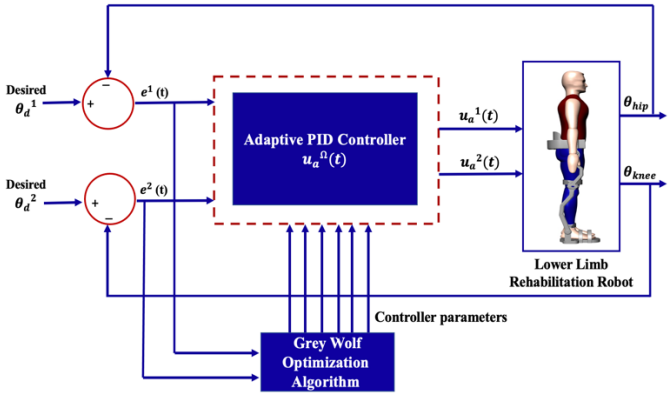


Figure 2 The block diagram of the APID

Sigma function is defined as stated in Eq. (6):

$$\sigma^{\Omega}(t) = X_P{}^{\Omega} e^{\Omega}(t) + X_D{}^{\Omega} e^{\cdot\Omega}(t) \qquad (6)$$

where Ω=1, 2 is the link number. e^{Ω} is the instantaneous error which represents the difference between the current desired trajectory $\theta_d{}^{\Omega}$ and actual output θ^{Ω} of link (Ω) as in Eq. (7):

$$e^{\Omega} = \theta_d{}^{\Omega} - \theta^{\Omega} \qquad (7)$$

The control law for this controller is given by Eq. (8):

$$u_a{}^{\Omega}(t) = M(\theta, \dot{\theta}) * u_{pid}{}^{\Omega}(t) \qquad (8)$$

Also, $u_{pid}{}^{\Omega}(t)$ is defined in Eq. (9):

$$u_{pid}{}^{\Omega}(t) = K_p{}^{\Omega}(t)\, e^{\Omega}(t) + K_i{}^{\Omega}(t) \int_0^t e^{\Omega}(t)\, dt + K_d{}^{\Omega}(t)\, \dot{e}^{\Omega}(t) \qquad (9)$$

where$K_p{}^{\Omega}(t), K_i{}^{\Omega}(t), and\ K_d{}^{\Omega}(t)$ are self-tune parameters obtained by Eq. (10) through Eq. (12):

$$K_p{}^{\Omega}(t) = \int \dot{K}_p{}^{\Omega}(t) \Longrightarrow \dot{K}_p{}^{\Omega}(t) = -\eta_1{}^{\Omega}\, \sigma^{\Omega}(t)\, e^{\Omega}(t) \qquad (10)$$

$$K_i{}^{\Omega}(t) = \int \dot{K}_i{}^{\Omega}(t) \Longrightarrow \dot{K}_i{}^{\Omega}(t) = -\eta_2{}^{\Omega}\, \sigma^{\Omega}(t) \int e^{\Omega}(t)\, dt \qquad (11)$$

$$K_d{}^{\Omega}(t) = \int \dot{K}_d{}^{\Omega}(t) \Longrightarrow \dot{K}_d{}^{\Omega}(t) = -\eta_3{}^{\Omega}\, \sigma^{\Omega}(t)\, \dot{e}^{\Omega}(t) \qquad (12)$$

where $\eta_1{}^{\Omega}$, $\eta_2{}^{\Omega}$, and $\eta_3{}^{\Omega}$ represent positive learning rate. It is crucial to choose the appropriate learning rates and initial values for the controller gains.

The optimal parameters of the APID controller $u_a(t)$ of link1 ($\eta_1{}^1, \eta_2{}^1,\ \eta_3{}^1, X_P{}^1,$ and $X_D{}^1$), and link2 ($\eta_1{}^2, \eta_2{}^2,\ \eta_3{}^2, X_P{}^2,$ and $X_D{}^2$) will be determined by the GWO algorithm.

Modified Adaptive PID (MAPID) Controller Design

In order to enhance the efficiency of the adaptive controller and improve its performance, a modified adaptive PID controller is suggested to reduce the steady state error, and overshot. Hence, the sigma function in Eq. (6) is modified as in Eq. (13):

$$\sigma^{\Omega}(t) = (X_P{}^{\Omega} e^{\Omega}(t) + X_I{}^{\Omega} \int e^{\Omega}(t) + X_D{}^{\Omega} e^{\cdot\Omega}(t) \qquad (13)$$

As such, the control law in Eq. (8) is modified to that in Eq. (14):

$$u_{ma}{}^{\Omega}(t) = M\big(\theta, \dot{\theta}\big)(u_{pid}{}^{\Omega}(t) + u_x{}^{\Omega}(t)) \qquad (14)$$

where $u_x{}^{\Omega}(t)$ is defined in Eq. (15):

$$u_x{}^{\Omega}(t) = tanh\,(\sigma^{\Omega}(t)) * K_M{}^{\Omega} \qquad (15)$$

To this end, the optimization of the parameters of the modified adaptive PID controller is accomplished by using GWO algorithm. The control goals are achieved through the fitness function design of this algorithm. The Lyapunov function candidate is set to be as in Eq. (16) and Eq. (17):

$$V^{\Omega} = \tfrac{1}{2}(\sigma^{\Omega}(t))^2 \qquad (16)$$

$$\dot{V}^{\Omega} = \sigma^{\Omega}(t)\dot{\sigma}^{\Omega}(t) < 0 \qquad (17)$$

when $\dot{V}^{\Omega} < 0$ it is guaranteed that $\sigma^{\Omega} \to 0$ as $t \to \infty$. From Eq. (13), it can be written:

$$\dot{\sigma}^{\Omega}(t) = (X_P{}^{\Omega} e^{\cdot\Omega}(t) + X_I{}^{\Omega} \int e^{\cdot\Omega}(t)\, dt + X_D{}^{\Omega} \ddot{e}^{\Omega}(t)) \qquad (18)$$

Substituting Eq. (18) into Eq. (17):

$$\dot{V} = \sigma^{\Omega}(t)[(X_P{}^{\Omega} e^{\cdot\Omega}(t) + X_I{}^{\Omega} \int \dot{e}^{\Omega}(t)\, dt + X_D{}^{\Omega} \ddot{e}^{\Omega}(t))\,] < 0 \quad (19)$$

The optimal parameters of the controller $u_{ma}(t)$ of link1 ($\eta_1{}^1, \eta_2{}^1, \eta_3{}^1, X_P{}^1, X_I{}^1, X_D{}^1$and $K_M{}^1$), and link2 ($\eta_1{}^2, \eta_2{}^2, \eta_3{}^2, X_P{}^2, X_I{}^2, X_D{}^2$ and $K_M{}^2$) are also determine by GWO algorithm, this is described in the next section.

Optimization Algorithm

Optimization is defined as the process of identifying the best solution for a specific problem to obtain the desired cost function properties to reach the global optimum. The optimization algorithms use multiple agents (solutions), to move through the search space in the process of solving an optimization problem. In this paper we use GWO technique (Grey Wolf Optimization Algorithm). The GWO fitness function ITAE (Integral Time Absolute Errors) is given by Eq. (20) for each rehabilitation robot link (1, 2):

$$F = ITAE = \int_0^{\infty} t|e(t)|dt \quad (20)$$

Grey Wolf Optimization Algorithm (GWOA)

Grey Wolf Optimization (GWO) is a population-based meta-heuristics algorithm proposed by Mirjalili et al. in 2014, it simulates grey wolf's leadership hierarchy and hunting strategy (Mirjalili, Mohammad and Lewis, 2014). Grey wolves are classified as apex predators. They prefer to live in packs, each pack consists of five to twelve members, and each member of the group adheres to a very strict social dominant hierarchy as illustrated in the Fig. (5) below.

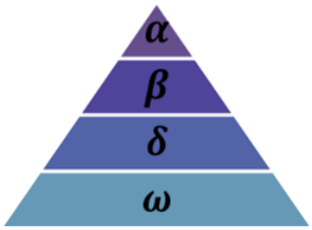


Figure 5 Social hierarchy of grey wolf (Mirjalili, Mohammad and Lewis, 2014)

This hierarchy is divided into four levels: the first is named Alpha (α), and the alpha wolves are the pack's leaders. They have the responsibility to decide on hunting, walking time, sleeping place etc. The pack members are required to obey the decision of alpha. The second level is known as Beta (β), and betas are subordinate wolves who assist the alpha in making decisions, the beta wolf is considered the best nominee to be the next alpha. They reinforce

the alpha's orders throughout the pack and provides feedback to the alpha. Delta (δ) is the third level which are not alpha or beta wolves, and they are referred to as subordinates. While the delta's wolves must submit to the alpha and beta, they dominate the omega (the lowest level in the social hierarchy of wolves). The fourth (lowest) level is known as Omega (ω). Omega wolves are the pack's scapegoats, and they must subordinate to all other dominant wolves. They may appear to be unimportant members of the pack, and they are the last wolves allowed to eat. In GWO algorithm, the fittest solution is considered the alpha, the second and the third fittest solutions are named beta and delta, respectively, the rest of the solutions are considered omega. The optimization is guided by α, and β, and δ might participate occasionally, the ω solutions follow these three wolves/agents. During the hunting the grey wolves encircle prey, the mathematical model of the encircling behaviour is presented in Eq. (21) and Eq.(22):

$$D_G = \left|C_G . X_{pG}(t) - A_G . X_G(t)\right| \quad (21)$$

$$X_G(t+1) = X_{pG}(t) - A_G . D_G \quad (22)$$

where t is the current iteration, A_G and C_G are coefficient vectors, X_{pG} is the position vector of the prey, and X_G indicates the position vector of a grey wolf.

The vectors $A_G$and C_G are calculated as in Eq. (23) and Eq. (24):

$$A_G = 2a_G . r_{1G} . a_G \quad (23)$$

$$C_G = 2 . r_{2G} \quad (24)$$

where components of a_G are linearly decreased from 2 to 0 over the course of iterations and r_{1G}, r_{2G} are random vectors in [0,1]. It is assumed that alpha, beta and delta have better knowledge about the potential location of pray, thus, the first three best solutions are saved and the other agent are obliged to update their positions according to the position of the best search agents as shown in Eq. (25) through Eq. (31):

$$D_{\propto G} = |C_{1G} . X_{\propto G} - X_G| a_G \quad (25)$$

$$D_{\beta G} = \left|C_{2G} . X_{\beta G} - X_G\right| a_G \quad (26)$$

$$D_{\delta G} = |C_{3G} . X_{\delta G} - X_G| \, a_G \quad (27)$$

$$X_{1G} = X_{\propto G} - A_{1G} . (D_{\propto G}) a_G \quad (28)$$

$$X_{2G} = X_{\beta G} - A_{2G} . \left(D_{\beta G}\right) a_G \quad (29)$$

$$X_{3G} = X_{\delta G} - A_{3G} . (D_{\delta G}) a_G \quad (30)$$

$$X_G(t+1) = \frac{X_{1G} + X_{2G} + X_{3G}}{3} \; a_G \quad (31)$$

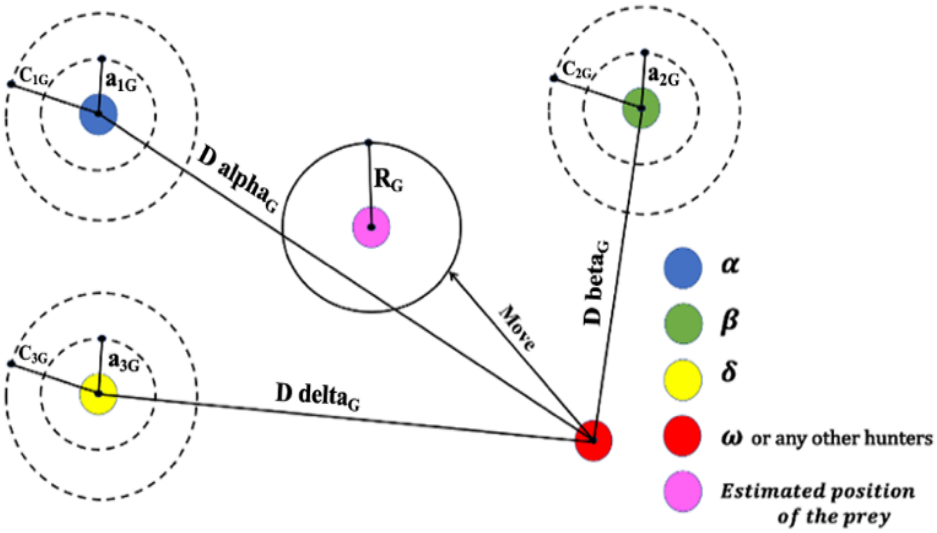


Figure 6 Position updating in GWO (Mirjalili, Mohammad and Lewis, 2014)

when $|A_G| < 1$, the wolves approach towards the prey which represents an exploration process. This process in GWO is applied according to the position, also, wolves diverge from each other to search for prey and converge for attack. The exploration process is modelled mathematically by utilizing A_G with random values greater than 1 or less than -1 (, $|A_G| > 1$) to oblige the search agents to diverge from the prey to find a fitter one. GWO finishes search by attacking the prey when it stops moving [(Mirjalili, Mohammad and Lewis, 2014) - (Muro *et al.*, 2011)].

Simulation Results

With the aid of the facility included in the MATLAB software version (R2019b), various lower limb rehabilitation robot simulations are performed for linear and nonlinear desired trajectories to demonstrate the efficiency of the APID and MAPID based on GWO algorithm. The GWO parameters for each rehabilitation robot link (1, 2) are given in Table 2, and final optimal parameters for APID and MAPID for link 1 and 2 are given in Table 3.

Table 2 The parameters of GWO algorithm

Parameters	APID	MAPID
No. of iterations	50	50
No. of search agents	30	30
dim (number of variables)	10	14
lower bound of variable n (lb)	[7; 7; 1; 7; 0.5; 7; 7; 1; 7; 0.5]	[7; 7; 1; 7; 0.5; 0.1; 7; 7; 1; 7; 0.5; 0.005; 77; 97]
upper bound of variable n (ub)	[13; 13; 3; 13; 2.5; 13; 13; 4; 13; 2.5]	[13; 13; 3; 13; 2.5; 1.5; 13; 13; 4; 13; 2.5; 0.05; 83; 103]

Table 3 Optimal parameters of the APID and MAPID obtained by GWO algorithm

Links	Controller parameters	Value	
		APID	MAPID
Link1 (hip)	$\eta_1{}^1$	9.80354	9.167085
	$\eta_2{}^1$	11.8295	13.00000
	$\eta_3{}^1$	2.77214	1.036600
	$X_P{}^1$	10.2836	13.00000
	$X_I{}^1$	/	1.201148
	$X_D{}^1$	2.22802	0.934094
	$K_M{}^1$	/	83.00000
Link2 (knee)	$\eta_1{}^2$	7.00000	10.87779
	$\eta_2{}^2$	13.0000	13.00000
	$\eta_3{}^2$	1.21741	3.096483
	$X_P{}^2$	7.15899	13.00000
	$X_I{}^2$	/	2.049219
	$X_D{}^2$	0.509149	0.013300
	$K_M{}^2$	/	103.0000

Table 4 The evaluation parameters of the APID and MAPID simulation results

Links	Parameter	Value	
		APID	MAPID
Link1 (hip)	$M_p(\%)$	0.056	0
	$t_s\,(sec.)$	2.985	0.463
	$e_{s.s}$	0.0000035	0.001
	$t_r\,(sec.)$	4.180	0.485
Link2 (knee)	$M_p(\%)$	-1.106	0
	$t_s\,(sec.)$	2.976	0.851
	$e_{s.s}$	0.00001	0.010
	$t_r\,(sec.)$	2.271	0.752

Simulation Results of the Linear Trajectory

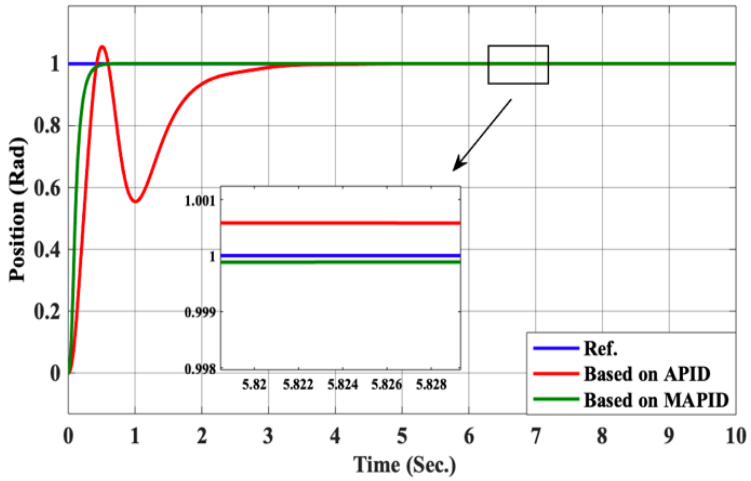
The step response (positive unity step for link1, and negative unity step for link2) of the controlled lower limb rehabilitation robot (position and control signal) are depicted in Fig.7 and Fig.8 for adaptive PID controller and modified adaptive PID controller. The results with APID demonstrate that the robot's performance is insufficient compared to the performance of MAPID which proves more efficient, where the rehabilitation robot follows the desired path very fast, with no overshoot, small steady-state trajectory error, and a smooth control signal (less than 400 N for link1 and approximately 100 N for link2). The evaluation parameters of simulation results for the APID and the MAPID controller are listed in Table 4. The performance of the simulation results are verified for link1 and link2

by the values of ITAE (Integral Time Absolute Error) and IAE (Integral Absolute Error) in Table 5.

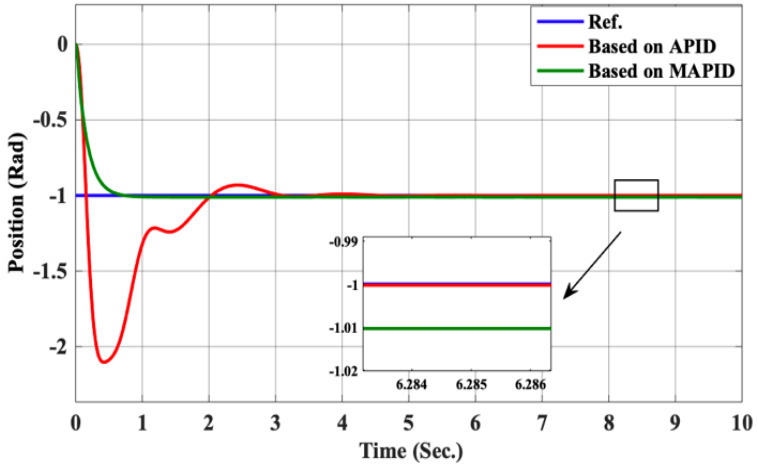
Table 5 The ITAE and IAE values of the APID and M APID for the linear trajectory

	Links	Value	
		APID	MAPID
ITAE	Link1	5.93	1.303
	Link2	9.847	2.563
IAE	Link1	0.593	0.1303
	Link2	0.9847	0.2563

We notice from Table 5 the values of ITAE and IAE for MAPID with linear trajectory is smaller than values for APID, this proves the MAPID controller is better than APID controller.

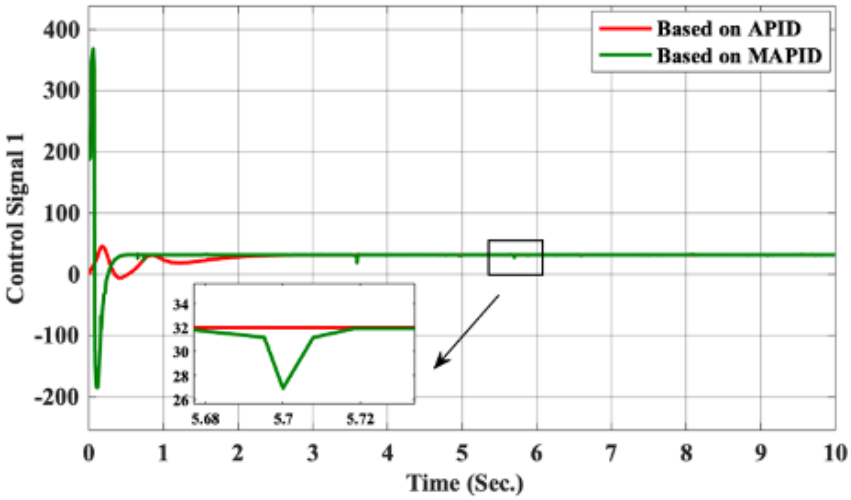


a. Link of hip Output

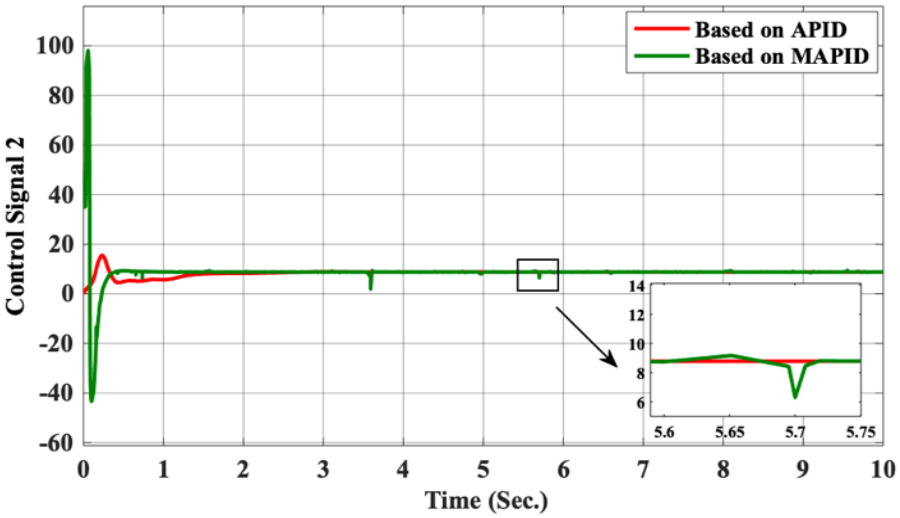


b. Link of knee output

Figure 7 The position of hip and knee links for linear trajectory with APID and MAPID



a. The control signal of hip link



b. The control signal of knee link

Figure 8 The control signals for linear trajectory with APID and MAPID

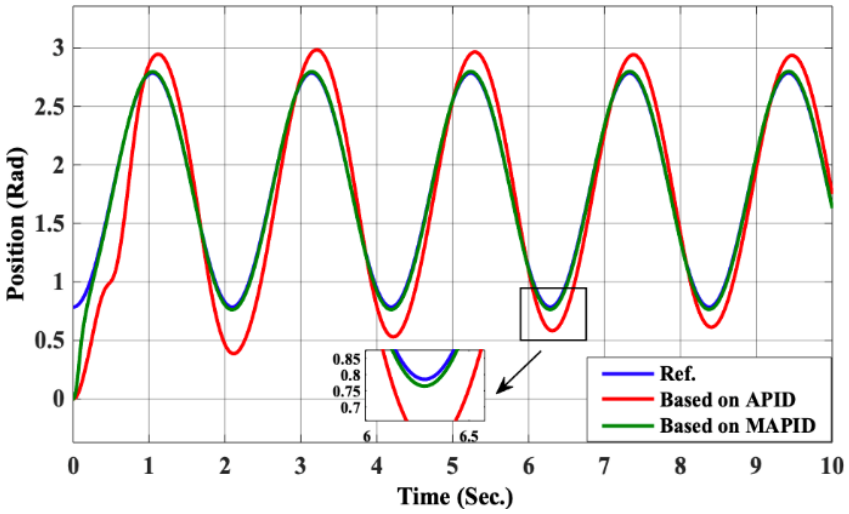
Simulation Results of the Non-Linear Trajectory

The simulation results of the lower limb rehabilitation robot with the APID and MAPID tested by the nonlinear cosine input signal ($x_d{}^1 = {}^{\pi}/_{4} + (1 - cos3t)$) for link1 and ($x_d{}^2 = {}^{\pi}/_{6} + (1 - cos5t)$) for link2 are illustrated in Fig. 9, and Fig. 10; these results show that despite the nonlinearity of the input signal, the performance is dependable. The results show the control signal is smooth (less than 400 N for link1 and less than 100 N for link2) with MAPID and the results show a bad performance with APID. Table 6 gives the values of ITAE and IAE for this experiment.

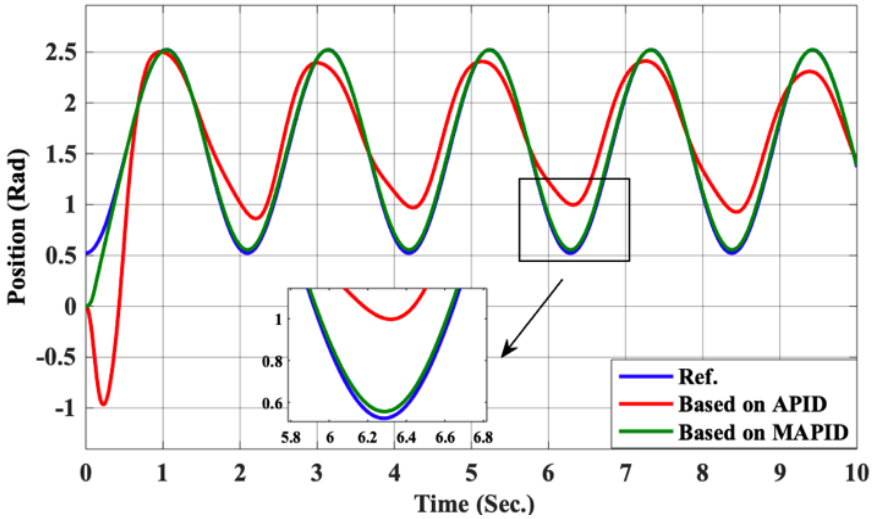
Table 6 The ITAE and IAE values of the APID and MAPID for the non-linear trajectory

	Links	Value	
		APID	MAPID
ITAE	Link1	20.44	2.145
	Link2	26.73	2.808
IAE	Link1	2.044	0.2145
	Link2	2.673	0.2808

We notice from Table 6 the values of ITAE and IAE for MAPID with non-linear trajectory is smaller than values for APID, this proves the MAPID controller is better than APID controller.

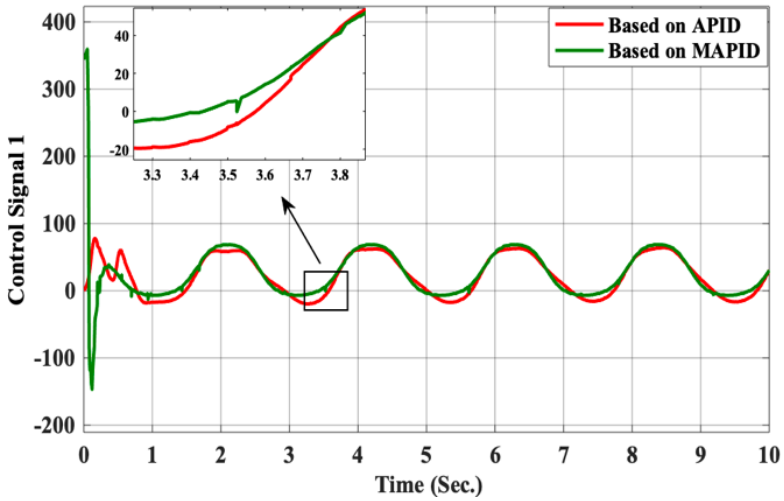


a. Link of hip output

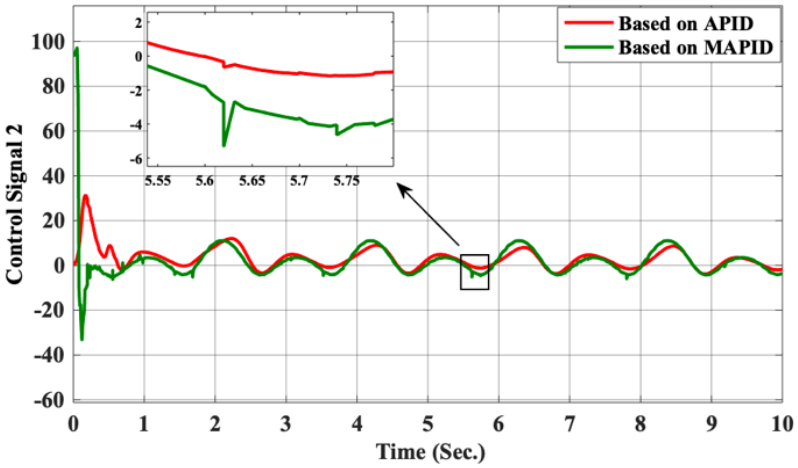


b. Link of knee output

Figure 9 The position of hip and knee links for non-linear trajectory with APID and MAPID



a. The control signals of hip link



b. The control signals of knee link

Figure 10 The control signals for non-linear trajectory with APID and MAPID

Conclusions

This work is dedicated to the investigation of the design of adaptive PID (APID) controller and suggest a modified adaptive PID (MAPID) controller for 2-DOF lower limb rehabilitation robot based on grey wolf optimization (GWO) algorithm to improve the performance of lower limb rehabilitation robot and track the desired trajectory. The overall results show the robot's performance is insufficient in the case of APID compared to the performance when using the MAPID controller. The results show that overshoot is reduced from 0.950 to 0 and from -1.929 to 0 respectively for link1 and link2, settling time is reduced from 2.985 to 0.463 and from 2.976 to 0.851 respectively for link1 and link2, has improved the robot's performance with no overshoot, small steady-state error, and a smooth control signal.

Conflicts of Interest

The author declare no conflict of interest.

Acknowledgment

Praise and thanks be to God who made all of this possible.

The author wants to share the deep appreciation for the excellent advice, motivation, and assistance provided to her supervisors, Professor Dr. Ekhlas H. Karam and Assistant Professor Dr. Muayed S AL-Huseiny.

References

Alshatti, A. (2019). *Design and Control of Lower Limb Assistive Exoskeleton for Hemiplegia Mobility*. (Doctoral dissertation, University of Sheffield). http://etheses.whiterose.ac.uk/24891/

Bien, Z., Chung, M.J., Chang, P.H., Kwon, D.S., Kim, D.J., Han, J.S., & Lim, S.C. (2004). Integration of a rehabilitation robotic system (KARES II) with human-friendly man-machine interaction units. *Autonomous robots, 16*(2), 165-191. http://doi.org/10.1023/B:AURO.0000016864.12513.77

Ebel, K.C. (2011). *Adaptive Sliding Mode Control for Aircraft Engines.* http://engagedscholarship.csuohio.edu/etdarchive/641

Gilbert, M., Zhang, X., & Yin, G. (2016). Modeling and design on control system of lower limb rehabilitation exoskeleton robot. In 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 348-352. http://doi.org/10.1109/URAI.2016.7734058

Ju, M.S., Lin, C.C., Lin, D.H., Hwang, I.S., & Chen, S.M. (2005). A rehabilitation robot with force-position hybrid fuzzy controller: hybrid fuzzy control of rehabilitation robot. *IEEE transactions on neural systems and rehabilitation engineering, 13*(3), 349-358. http://doi.org/10.1109/TNSRE.2005.847354

Mech, L.D. (1999). *Alpha status, dominance, and division of labor in wolf packs.* https://digitalcommons.unl.edu/usgsnpwrc/353.

Mirjalili, S., Mirjalili, S.M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software, 69*, 46-61. http://doi.org/10.1016/j.advengsoft.2013.12.007

Muro, C., Escobedo, R., Spector, L., & Coppinger, R.P. (2011). Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations. *Behavioural processes, 88*(3), 192-197. http://doi.org/10.1016/j.beproc.2011.09.006

Alireza, R., Bahman, M., & Waiter, B. (2005). Adaptive PID controller using PC matlab. *In Proceedings of the American Society for Engineering Education Annual Conference & Exposition, Portland, Oregon, June*, 12-15. http://doi.org/10.18260/1-2--15038

Al Rezage, G., & Tokhi, M.O. (2016). Fuzzy PID control of lower limb exoskeleton for elderly mobility. In IEEE International Conference on Automation, *Quality and Testing, Robotics (AQTR)*, 1-6. http://doi.org/10.1109/AQTR.2016.7501310

Saryanto, W.Y., Cahyadi, A.I., & Herianto (2014). Design and Development of Knee and Ankle Rehabilitation Robot Based on Indonesian Anthropometric Data, 413–416.

Torabi, M., Sharifi, M., & Vossoughi, G. (2018). Robust adaptive sliding mode admittance control of exoskeleton rehabilitation robots. *Scientia Iranica*, *25*(5), 2628-2642. http://doi.org/10.24200/sci.2017.4512

Veneman, J.F., Kruidhof, R., Hekman, E.E., Ekkelenkamp, R., Van Asseldonk, E.H., & Van Der Kooij, H. (2007). Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 15*(3), 379-386. http://doi.org/10.1109/TNSRE.2007.903919

Winter, D.A. (2009). *Biomechanics and motor control of human movement.* John Wiley & Sons.

Wu, J., Huang, J., Wang, Y., & Xing, K. (2012). RLSESN-based PID adaptive control for a novel wearable rehabilitation robotic hand driven by PM-TS actuators. *International Journal of Intelligent Computing and Cybernetics. 5*(1), 91–110. http://doi.org/10.1108/17563781211208242

Zahid, J., Khor, K.X., Yeong, C.F., Su, E.L. M., & Duan, F. (2017). Adaptive control of DC motor for one-DOF rehabilitation robot. *ELEKTRIKA-Journal of Electrical Engineering, 16*(3), 1-5. http://doi.org/10.11113/elektrika.v16n3.29