Two-link lower limb exoskeleton model control enhancement using computed torque

Elang Parikesit, Dechrit Maneetham
Department of Mechatronics Engineering, Rajamangala University of Technology Thanyaburi, Pathum Thani, Thailand

Article Info

Article history:

Received Feb 28, 2023
Revised May 16, 2023
Accepted Jun 26, 2023

Keywords:

Computed torque controller
Exoskeleton
Gait trainer
Lower limb
Robotics

ABSTRACT

Robotic technology has recently been used to help stroke patients with gait and balance rehabilitation. Rehabilitation robots such as gait trainers are designed to assist patients in systematic, repetitive training sessions to speed up their recovery from injuries. Several control algorithms are commonly used on exoskeletons, such as proportional, integral and derivative (PID) as linear control. However, linear control has several disadvantages when applied to the exoskeleton, which has the problem of uncertainties such as load and stiffness variations of the patient's lower limb. To improve the lower limb exoskeleton for the gait trainer, the computed torque controller (CTC) is introduced as a control approach in this study. When the dynamic properties of the system are only partially known, the computed torque controller is an essential nonlinear controller. A mathematical model forms the foundation of this controller. The suggested control approach's effectiveness is evaluated using a model or scaled-down variation of the method. The performance of the suggested calculated torque control technique is then evaluated and contrasted with that of the PID controller. Because of this, the PID controller's steady-state error in the downward direction can reach 5.6%, but the CTC can lower it to 2.125%.

Corresponding Author:

Dechrit Maneetham
Department of Mechatronics Engineering, Rajamangala University of Technology Thanyaburi
Tambon Khlong Hok, Amphoe Khlong Luang, Pathum Thani, Thailand
Email: dechrit_m@rmutt.ac.th

1. INTRODUCTION

One of the leading causes of death and acquired long-term disability globally, the likelihood of stroke increases with aging [1]–[5]. Mobility issues brought on by stroke or other related illnesses are becoming more common [6]. When one cannot walk normally, they depend on wheelchairs or other mobility devices like orthotics or ankle braces. For many recovering from neurological illnesses, regaining the ability to walk is one of the key objectives. Robotic devices have been utilized in various treatments to help people retrain their motor functions and walk again [7], [8]. Robotic gait/walking trainers including the G-EO, Gait Trainer GT II, Lexo, Lokomat®, and Walkbot are available through market vendors, but the cost is still prohibitive [9]. Innovation that can "do more with less" is important to provide significantly more commercial and social gain while using fewer finite resources like electricity, money, and time [10].

The robotic exoskeleton has been briefly described in several recent studies [11]–[15]. Gait trainers, also known as lower limb rehabilitation exoskeleton robots, are an important class of rehabilitation robots that can control every joint's motion during training by wearing a wearable connection to the user's body. Exoskeleton robots for healing lower limbs were the focus of research in the 1960s [16], [17]. Some researchers use proportional, integral, and derivative (PID) as controllers of these devices [18]–[20] because it is the most popular controller due to its high dependability and straightforward installation due to its great durability and simplicity.

Despite their adaptability, current PID controllers have disadvantages that limit their performance, especially regarding robustness, disturbance rejection and tracking. PID controller alternatives that offer higher performance at a reasonable cost include fuzzy control, generalized linear control, and observer and state feedback. These alternatives frequently necessitate a thorough understanding of control theory, in addition to being more challenging to configure than a straightforward PID controller [21]. Munadi *et al.* [22] researched that exoskeleton using PID as a control. The PID controller can achieve steady-state conditions at the set point, but there are still overshoots.

Many controllers for position control of exoskeleton robots are linear, which may be better for exoskeleton robots with complex structures and high levels of uncertainty [23]. Because of factors like changes in load, friction, and outside interference, for an exoskeleton robot, a PID-only controller is insufficient [24], [25]. Even if separate PID controls are sufficient for many set-point regulation issues, many tasks call for advanced trajectory-tracking tools. There are several suggestions to overcome these PID controller weaknesses. The first is to handle the non-linearity using feedback linearization [26]. Second, improving trajectory tracking performance by employing a computed-torque control strategy that takes into account the manipulator’s dynamic model [27].

There has been prior study on planar two-link exoskeletons and robotic manipulators that use PID, model predictive control (MPC), and linear quadratic (LQ) optimal control. PID control for a two-link arm manipulator was proposed by Robles [28] proposed, while the techniques proposed by Guechi *et al.* [29] (MPC control and LQ control) outperform the PID in terms of system performance. The contribution of this paper is a two-link lower limb exoskeleton model that can overcome the problem of uncertainties such as load and stiffness variations of the patient’s lower limb.

2. METHOD

Figure 1 shows the general layout of a typical gait rehabilitation control system [30]. The rehabilitation system hardware and multi-level control architecture can be divided into two main groups: physiotherapy algorithms (high level) and feedback control algorithms (low level). This research will focus on the low-level hardware and controller, major in the lower limb exoskeleton’s overall performance.

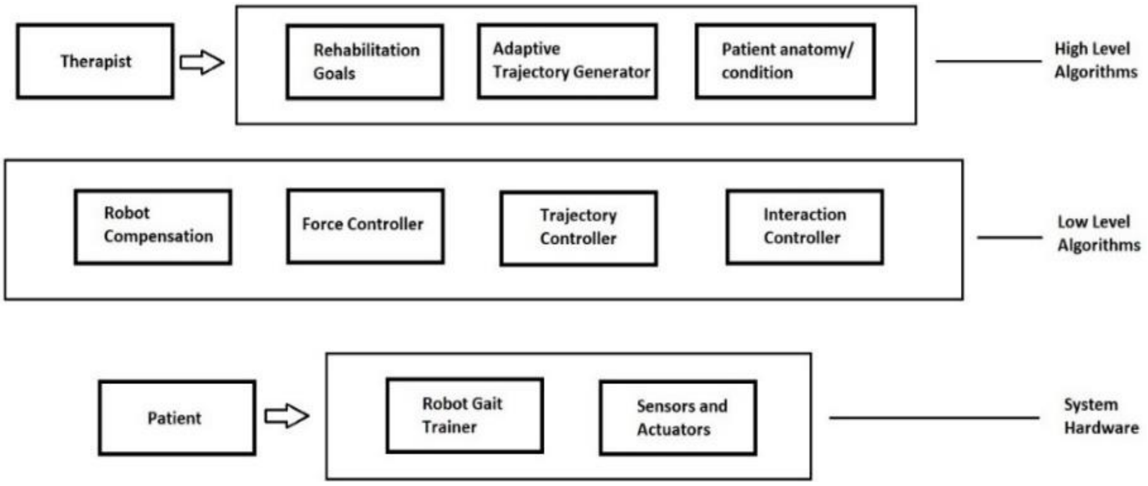


Figure 1. The architecture of the gait rehabilitation control system

2.1. Kinematics and dynamics

The kinematics and dynamics of the system must be taken into consideration while constructing a control system. Figure 2(a) shows the Lokomat®, a robotic gait trainer product from Hocoma [31]. While Figure 2(b) shows the low-cost version of the gait trainer built by the authors. Figure 2(c) depicts the kinematics of the two-link manipulator.

2.1.1. Kinematics

The exoskeleton for the lower limbs is identical to a planar RR arm with two links. Refer to Figure 2 to determine it is dynamics, where the masses of the link are assumed to be concentrated at it is ends [32]. Parameter of joint is (1).

$$q = [\theta_1 \quad \theta_2] \tag{1}$$

Together with the total vector of force being

$$\tau = [\tau_1 \quad \tau_2]T \tag{2}$$

with the actuators supplying τ_1 and τ_2 torque kinetic and potential energy for link 1 is

$$K_1 = {}^1\!/_2\, m_1 a_1^2 \dot{\theta}_1^2 \tag{3}$$

$$P_1 = m_1 g a_1 sin\theta_1 \tag{4}$$

for link 2 we have

$$X_2 = a_1 \cos\theta_1 + a_2 \cos(\theta_{1+}\theta_2) \tag{5}$$

$$Y_2 = a_1 \sin\theta_1 + a_2 \sin(\theta_{1+}\theta_2) \tag{6}$$

$$\dot{X}_2 = -a_1\dot{\theta} \sin\theta_1 - a_2 \cos(\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_{1+}\theta_2) \tag{7}$$

$$\dot{Y}_2 = -a_1\dot{\theta} \sin\theta_1 - a_2 \cos(\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_{1+}\theta_2) \tag{8}$$

for the velocity squared to be (9).

$$v_2^2 = \dot{X}_2^2 + \dot{Y}_2^2 = a_1^2\dot{\theta}_1^2 + a_2^2(\dot{\theta}_1 + \dot{\theta}_2)^2 + 2a_1a_2(\dot{\theta}_1 + \dot{\theta}_1\dot{\theta}_2)^2 \cos\theta_2 \tag{9}$$

This means that link 2’s kinetic energy equals.

$$K_2 = {}^1\!/_2\, m_2 v_2^2 = {}^1\!/_2\, m_2 a_2^2 \dot{\theta}_1^2 + {}^1\!/_2\, m_2 a_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2a_1a_2(\dot{\theta}_1 + \dot{\theta}_1\dot{\theta}_2) cos\theta_2 \tag{10}$$

In terms of potential energy stored in link 2:

$$P_2 = m_2 g y_2 = m_2 g[a_1 \sin\theta_1 + a_2 \sin(\theta_{1+}\theta_2)] \tag{11}$$

As a whole, the arm’s Lagrangian equation is as (12).

$$\begin{aligned} &L = K - P = K_1 + K_2 - P_1 - P_2 \\ &= {}^1\!/_2\,(m_1 + m_2) a_1^2 \dot{\theta}_1^2 + {}^1\!/_2\, m_2 a_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 a_1 a_2 (\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2) cos\theta_2 \\ &-(m_1 + m_2) g a_1 \sin\dot{\theta}_1 - m_2 g a_2 \sin(\theta_{1+}\theta_2) \end{aligned} \tag{12}$$

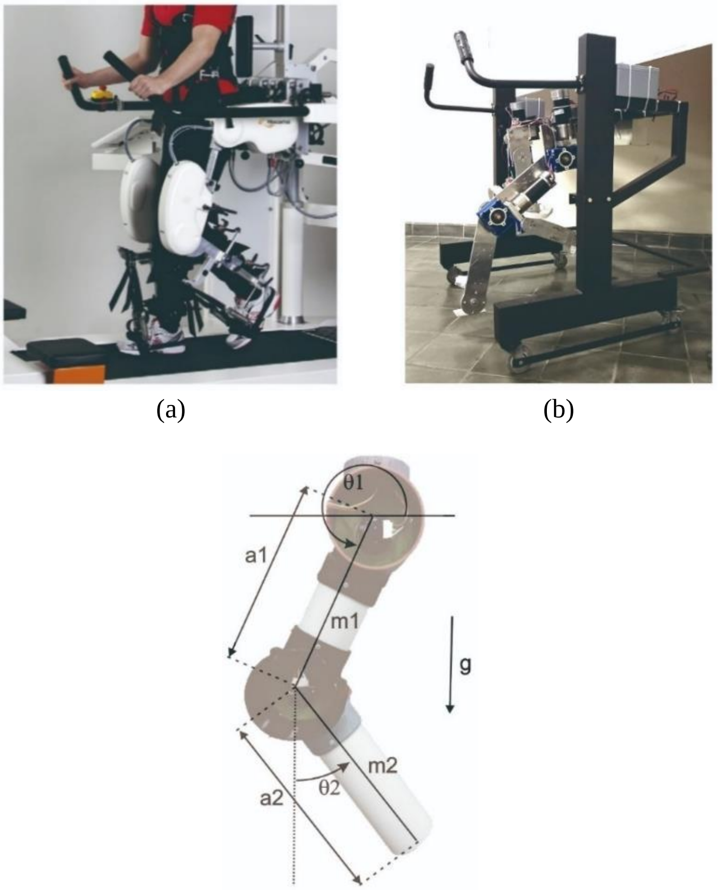
The terms needed for (14) are

$$\begin{aligned} &\frac{\partial L}{\partial \dot{\theta}_1} = (m_1 + m_2)\, a_1^2 \dot{\theta}_1^2 + m_2 a_2^2 (\dot{\theta}_1 + \dot{\theta}_2) + m_2 a_1 a_2 (2\dot{\theta}_1 + \dot{\theta}_2) cos\theta_2 \\ &\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_1} = (m_1 + m_2)\, a_1^2 \ddot{\theta}_1^2 + m_2 a_2^2 (\ddot{\theta}_1 + \ddot{\theta}) + m_2 a_1 a_2 (2\ddot{\theta}_1 + \dot{\theta}_2) cos\theta_2 \\ &-m_2 a_1 a_2 (2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) sin\theta_2 \\ &\frac{\partial L}{\partial \theta_1} = -(m_1 + m_2) g a_1 \cos\theta_1 - m_2 g a_2 cos(\theta_{1+}\theta_2) \\ &\frac{\partial L}{\partial \dot{\theta}_2} = m_2 a_2^2 (\dot{\theta}_1 + \dot{\theta}_2) + m_2 a_1 a_2 \dot{\theta}_1 cos\theta_2 \\ &\frac{d}{dt}\frac{\partial L}{\partial \theta_2} = m_2 a_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 a_1 a_2 \ddot{\theta}_1 cos\theta_2 - m_2 a_1 a_2 \dot{\theta}_1 \dot{\theta}_2 sin\theta_2 \\ &\frac{\partial L}{\partial \theta_2} = -m_2 a_1 a_2 (\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2) sin\theta_2 - m_2 g a_2 \cos(\theta_{1+}\theta_2) \end{aligned}$$

Finally, the lower limb dynamics are provided by coupled nonlinear differential equations according to Lagrange’s (13) and (14).

$$\tau_1 = [(m_1 + m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2cos\theta_2]\ddot{\theta}_1$$
$$+ [m_2a_2^2 + m_2a_1a_2cos\theta_2]\ddot{\theta}_2 - m_2a_1a_2(2\dot{}\theta_1\dot{\theta}_2 + \dot{\theta}_2^2)sin\theta_2$$
$$+(m_1 + m_2)ga_1cos\theta_1 + m_2ga_2cos(\theta_1 + \theta_2) \quad (13)$$

$$\tau_2 = [m_2a_2^2 + m_2a_1a_2cos\theta_2]\ddot{\theta}_1 + m_2a_2^2\ddot{\theta}_2 + m_2a_1a_2\dot{\theta}_1^2sin\theta_2$$
$$+ m_2ga_2cos(\theta_1 + \theta_2) \quad (14)$$



(a) (b)

(c)

Figure 2. Robotic gait trainers (a) Lokomat® from Hocoma, (b) the low-cost version of gait trainer, and (c) 2 links kinematics model

2.1.2. Dynamics

When the lower limb dynamics are written out in vector form, the result is

$$\begin{bmatrix} (m_1 + m_2)a_2^1 + m_2a_2^2 + 2m_2a_1a_2 \cos\theta_2 & m_2a_2^2 + m_2a_1a_2 \cos\theta_2 \\ m_2a_2^2 + m_2a_1a_2 \cos\theta_2 & m_2a_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$
$$+ \begin{bmatrix} -m_2a_1a_2 \left(2\,\dot{\theta}_1\,\dot{\theta}_2\right)\sin\dot{\theta}_2 \\ m_2a_1a_2\theta_2^1 \sin\theta_2 \end{bmatrix} + \begin{bmatrix} (m_1m_2)ga_1\cos\theta_1 + m_2\,ga_1\cos(\theta_1 + \theta_2) \\ m_2\,ga_1\cos(\theta_1 + \theta_2) \end{bmatrix}$$
$$= \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (15)$$

This manipulator dynamics is expressed in its typical manner.

$$M(q)\ddot{q} + V(q.\dot{q}) + G(q) = \tau_3 \quad (16)$$

where $M(q)$ is manipulator dynamic (symmetric matrix), $V(q,\dot{q})$ is vector of Coriolis/centripetal, G(q) is vector of gravity.

2.2. Control strategy of gait trainer

2.2.1. Selection of control algorithms

There are three types of control algorithms in the history of control theory: classical, modern, and intelligent. Traditional techniques such basic propositional derivative (PD) and PID linear control, while necessary, are ineffective for nonlinear systems. Motion control is an increasingly used type of modern control because it has a simple structure and superior control performance. The best performance can be achieved with intelligent control, but it is challenging to develop and requires much knowledge. Numerous innovative and sophisticated control strategies have lately been developed for nonlinear mechanical systems [33]. For nonlinear mechanical systems, several cutting-edge and contemporary control strategies have been developed, including adaptive control, computed torque control, fuzzy control, neural network control, and sliding mode control [34].

2.2.2. Model learning

Techniques for building models from data have become increasingly valuable robotics tools in recent years. This occurrence is due to a multitude of factors. Standard models, such as rigid body dynamics, are only approximate because of the sophistication of today’s robotic systems. This estimate does not accurately model unknown non-linear drive sources (hydraulic and motor saturation), passive elements such as cables, hydraulic hoses, cableways friction and friction sources [35]. Table 1 lists the various model types (forward, inverse, mixed, and operator) and model learning architectures.

Table 1. An overview of the model types and how they relate to the various learning architectures and sample applications [27]

Type of model	Architecture of learning	Example applications
Operator	Direct	Model predictive control Planning, Optimization, Compensation for delays
Forward	Direct	Filtering Learning simulation Optimization Prediction
Inverse	Direct, indirect	Controls for inverse dynamics Computed torque Feedback linearization
Mixed	Direct	Inverse kinematics
	Indirect	Control of operational space
	Distal teacher	Multiple control of model

2.2.3. Feedback linearization

In a control system, feedback linearization transforms a nonlinear plant-the process or system under control-into a linear plant. In a nonlinear system, this is primarily done to improve the effectiveness of a linear control strategy. However, compensating for nonlinearities and dynamic coupling must happen more quickly than the control speed. By including a linearizing and decoupling controller in the direct control loops, the feedback linearization technique (FLT) achieves this goal.

In the past, analog control circuit design hindered the creation of nonlinear functions from overcoming nonlinear generators due to escalating hardware complexity, which raises prices and reduces reliability [36]. A nonlinear and coupled mechanical dynamic system’s feedback linearization is described [37], such as a robotic manipulator. The term “model-based” refers to controllers built on dynamic models. Passivity-based and calculated torque controllers are two broad categories for model-based controllers [38].

2.2.4. Computed torque control

Gradually with time, a wide range of different control systems of a robot is being presented as options. Research from the 1970s gave rise to the computed torque controller, first proposed by Paul [39]. The subsequent research has led to problems with its actual implementation (such as computation complexity and incorrect models) [40]. Computed torque control is a method for generalizing feedback linearization to nonlinear systems. Known also as the inverse dynamics control, the computed torque control depends on the robot dynamics’ inversion [41]. Introducing the term servo, which takes the form of a PD controller, computed torque controls return a linearization to account for non-linearity in manipulator dynamics [41]. The block diagram of the computed torque control technique is shown in Figure 3 feedback linearization can be time- and resource-intensive, so it might be difficult to apply it to change a nonlinear system into a linear one globally.

Computed torque controllers provide robust performance characteristics, according to the experiment [42]–[46]. Numerous studies using the computed torque control algorithm have been completed successfully [47]–[49]. The disadvantage of computed torque controls, however, is the requirement for tracking-related real-time system dynamics computation [50]. Consider the control input as (17).

$$\tau \ = H(q)v + \dot{C}(q, \dot{q})\dot{q} + \tau_g(q) \tag{17}$$

This system, called torque calculation control, constituted of: i) an inner loop of correction for nonlinearity and ii) a second loop that receives an external control signal. This indicates that if these control principles are included in the robotic arm's dynamic model.

$$\ddot{q} = v$$

This control input transforms the complex challenge of designing a nonlinear controller into a simple issue of designing a linear system with n subsystems. PD feedback is one method for controlling the outer loop *v*.

$$v = \ \ddot{q}_d + K_V \dot{e}_q + \ K_p e_q \tag{18}$$

This makes the total controlling input into

$$\tau = \ H(q) + \left(\ddot{q}_d + K_V \dot{e}_q + \ K_p e_q\right) + C(q, \dot{q})\dot{q} + \tau_g(q) \tag{19}$$

which leads to linear error dynamics that are (20).

$$\ddot{e}_q + K_v \dot{e}_q + K_p e_q = 0 \tag{20}$$

The theory of linear system predicts that the error in tracking would eventually reach zero.

Computed torque control (CTC) for robot control systems is a robust motion control method that can provide global asymptotic stability [51]. However, a conventional CTC can only be used efficiently if a precise and thorough dynamic model of a manipulator is provided [52]. Shown in Table 2 are parameters of a two-link model.

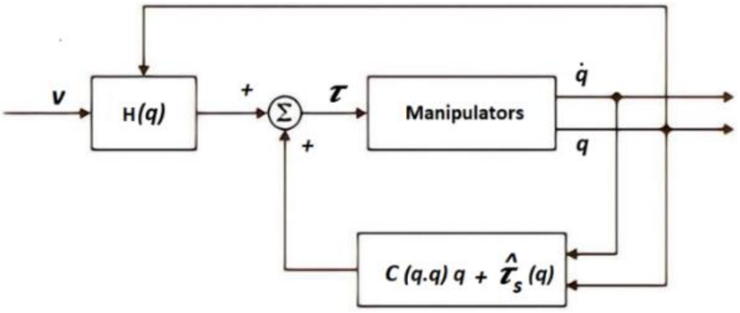


Figure 3. Computed-torque control scheme

Table 2. Two-link parameters

Parameters	Units	Values
Link 1 (L1)	m	0.2
Link 2 (L2)	m	0.215
L1 mass (m1)	kg	0.2
L2 mass (m2)	kg	0.2
Gravity (g)	m/s2	9.8

2.3. Microcontroller and motor driving circuit

As shown in Figure 4, two DC motors move the two links in the mechanism, while encoders serve as position sensors. The Arduino Uno microcontroller and the Monster Moto Shield VNH30SP DC motor driver circuit are used to control the motors. Based on the ATmega 328 microcontroller, Arduino is the primary controller. The Monster Moto Shield VNH30SP motor driver module will enhance the current to the motor.

With a sampling time of 10 ms, the encoder will provide data on the link’s current position. The microcontroller will use the computed torque control technique to determine the pulse-width modulation (PWM) value to send to the driving circuit.

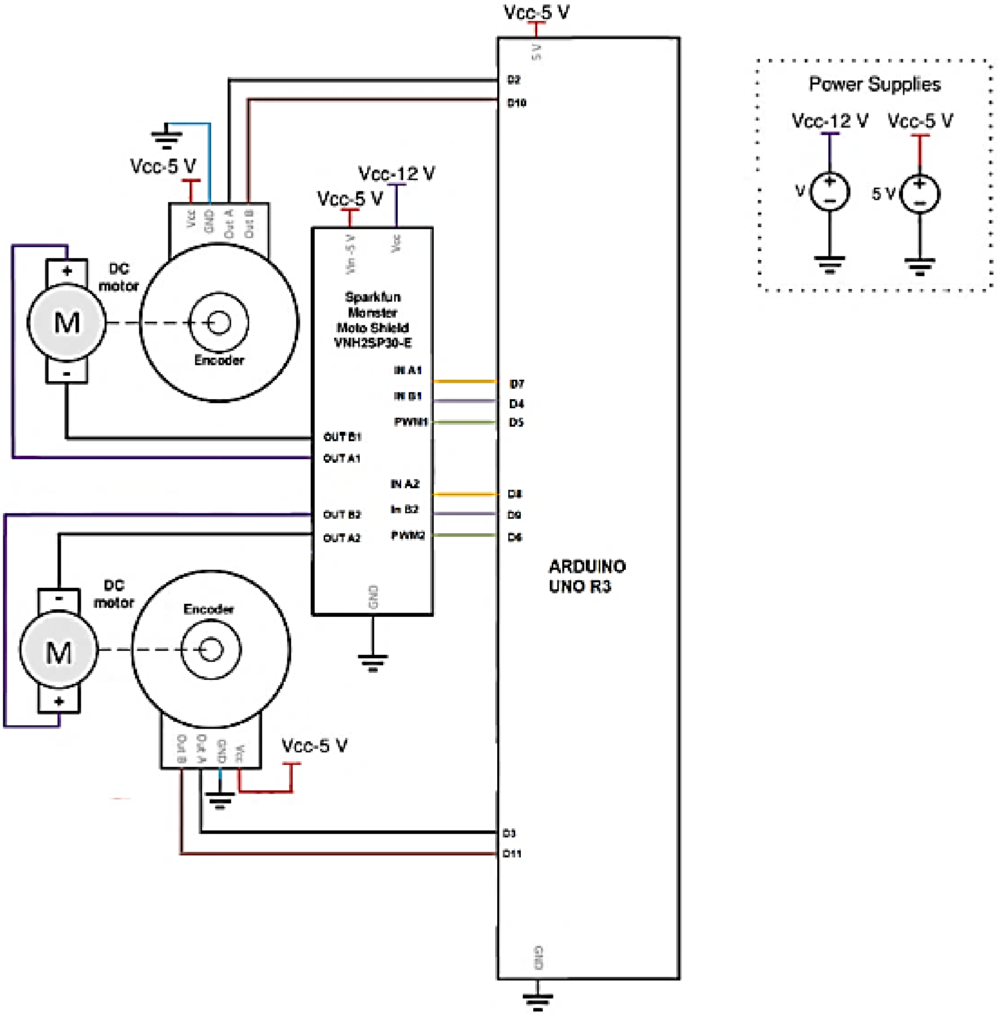


Figure 4. Wiring diagram of the experimental setup

3. RESULTS AND DISCUSSION

Figures 5 and 6 show the output position using the PID controller, while Figures 7 and 8 show the output position using the computed torque controller. The obtained results of both controllers are compared. It can be seen from all the graphs that both PID and CTC can suppress the overshoot. Both controllers have good performances for downward movements but need better performances for upward movements because there is additional gravity as the disturbance. Figure 5 shows the plot of the position of motor 1 and motor 2 while moving downward using the PID controller.

The parameters for motor 1 are: Kp=0.5; Ki=0.01; Kd=1.1, while for motor 2: Kp=0.5; Ki=0.01; Kd=1.1. The PID controller worked well in that situation. At motor 1 the rise time is 25 ms, with a steady-state error of 0.0167%. Meanwhile, at motor 2 the rise time is 19 ms, with a steady-state error of 0.75%.

Figure 6 shows the position of motor 1 and motor 2 while moving upward using the PID controller. The parameters for motor 1 are: Kp=0.5; Ki=0.01; Kd=1.1, while for motor 2: Kp=0.5; Ki=0.01; Kd=1.1. In this situation, the PID controller did not work well at motor 2, and there is a steady-state error of more than 5%. There is a gravity disturbance that the PID controller cannot handle. At motor 1 the rise time is 21 ms, with a steady-state error of 0.1%. Meanwhile, at motor 2 the rise time is 17 ms, with a steady state error of 5.6%.

Figure 7 shows the plot of the position of motor 1 and motor 2 while moving downward using the CTC controller. The parameters for motor 1 are: Kp =4; Ki=0.001; Kd=0.0005, while for motor 2: Kp=0.35;

Ki=0.001; Kd=0.0005. In this situation, the CTC controller worked well, even though at motor 2, there was still a little steady-state error. At motor 1 the rise time is 21 ms, with a steady state error of 1.04%. Meanwhile, at motor 2 the rise time is 30 ms, with a steady-state error of 2.175%.

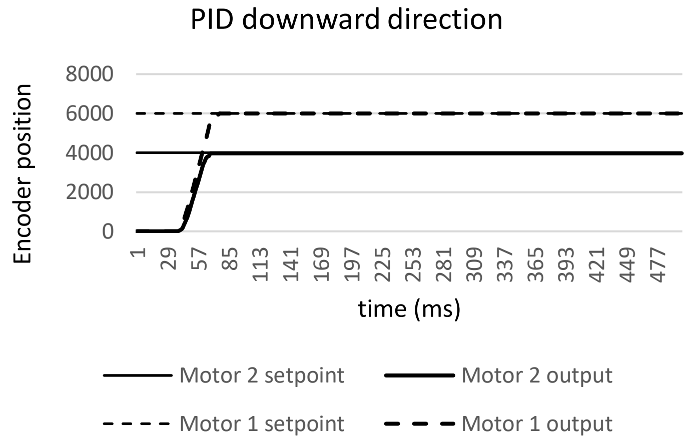


Figure 5. The plot of setpoint vs. PID output in the downward direction

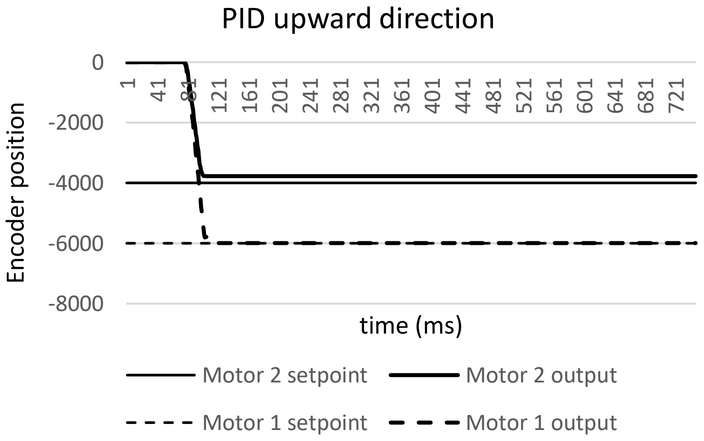


Figure 6. The plot of setpoint vs. PID output in the upward direction

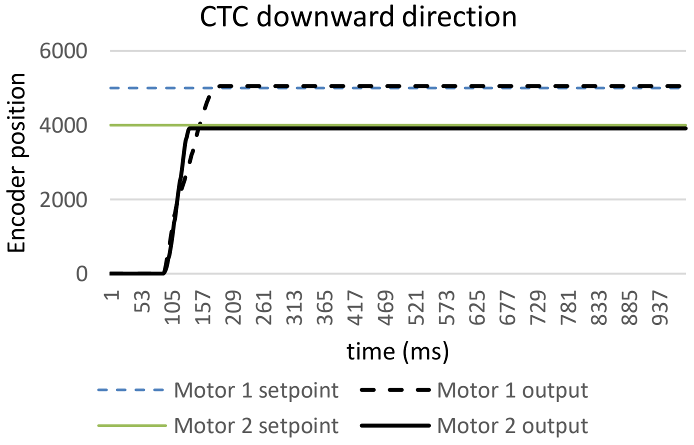


Figure 7. The plot of setpoint vs. CTC output in the downward direction

The plot of motor 1 and motor 2 positions during upward motion using CTC is shown in Figure 8. The parameters for motor 1 are: Kp=4; Ki=0.001; Kd=0.0005, while for motor 2: Kp=0.35; Ki=0.001; Kd=0.0005. CTC performed very well under these conditions. As mentioned, the computed torque controller method adjusts for gravity effects. There is a steady state inaccuracy of 1.74% with a rise time of 44 ms at motor 1. At the same time, motor 2 had a rise time of 25 ms and a steady-state error of 2.125%. The performance (rise time, steady state error, and overshoot) of CTC and PID controllers for downward and upward motion are summarized in Tables 3 and 4. Although CTC is better than PID controller in dealing with disturbance (gravity), PID controller has a faster rise time.

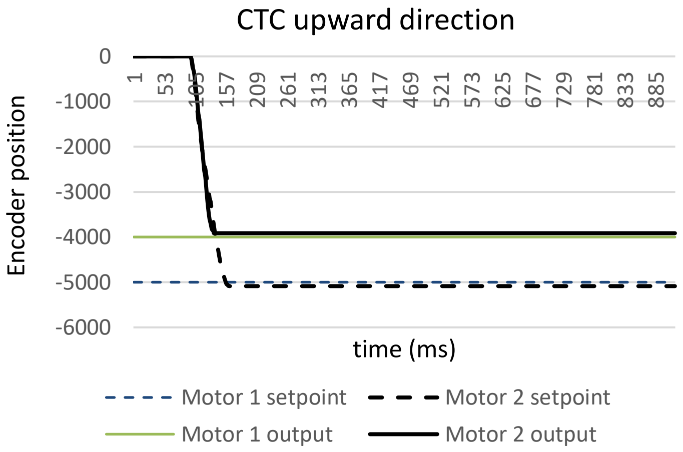


Figure 8. The plot of setpoint vs. CTC output in the upward direction

Table 3. Performance of PID controller in the downward and upward direction

		Rise time	Steady state error	Overshoot
PID in downward direction	M1	25 ms	0.0167%	-
	M2	19 ms	0.75%	-
PID in upward direction	M1	21 ms	0.1%	-
	M2	17 ms	5.6%	-

Table 4. Performance of CTC controller in the downward and upward direction

		Rise time	Steady state error	Overshoot
CTC in downward direction	M1	64 ms	1.04%	-
	M2	30 ms	2.175%	-
CTC in upward direction	M1	44 ms	1.74%	-
	M2	25 ms	2.125%	-

The system performance index can be calculated or assessed to assess system performance. The root mean square error (RMSE) and the mean square error (MSE) might both be used in this case. The acronym MSE means the square of RMSE. A statistic known as the RMSE contrasts actual values with predictions made by a hypothetical model [53]. MSE is represented as follows:

$$MSE = \frac{\sum_{j=1}^{V} E_J^2}{V}$$

V is quantity of data. The expression for RMSE is as follows:

$$RMSE = \sqrt{\frac{\sum_{j=1}^{V} E_J^2}{V}}$$

According to the RMSE and MSE, Tables 5 and 6 compare the PID and CTC responses numerically. It can be seen from the MSE and RMSE analysis that for upward movements, CTC always has better responses than PID. However, for the downward direction, the response of motor 2 (M2) using CTC is not as good as PID.

Table 5. Comparison of PID and CTC responses in an upward direction by numerical analysis

Joints	Control type	RMSE	MSE
M1	PID	35.83	861605.63
	CTC	31.22	895538.63
M2	PID	604.61	406298.42
	CTC	16.31	244404.95

Table 6. Comparison of PID and CTC responses in a downward direction by numerical analysis

Joints	Control type	RMSE	MSE
M1	PID	92.40	3944754.63
	CTC	54.44	2913817.40
M2	PID	61.11	1721801.91
	CTC	217.34	213643.10

4. CONCLUSION

This study examines the usage of a computed torque controller on a straightforward 2-link exoskeleton model for a robotic gait trainer, then compares its performance with a PID controller. While the lower limb's movement is non-linear, the PID controller is linear. So, we need another control that can work on a non-linear system. It is the computed torque controller in this instance.

In robotics and mechatronics, a method called computed torque control is employed to control the motion of a robotic system. Using the system's equations of motion, the needed torques at the joints of the robotic system are computed to create the desired motion. In the case of a two-link lower limb exoskeleton of a robotic gait trainer, the computed torque control method would involve calculating the appropriate torques at the joints using the equations of motion for the two links and the joints linking them to accomplish the desired motion of the lower limb.

The Lagrangian technique has been used to examine the two-link mechanism's dynamics and describe its kinematics. A model of 2 links mechanism model has also been created. This model of 2 links lower limb exoskeleton model will be used to improve a robotic gait trainer model. The PID controller's performance was compared to the CTC's in both upward and downward directions.

From the result, the suggested computed torque controller for this two-link model has advantages in dealing with disturbance. However, the PID controller is still faster than CTC for the rise time. The experiments show that both the PID controller and CTC can suppress overshoot. The PID controller cannot compensate for gravity as the disturbance, but CTC can do that. As a result, in the downward direction, the steady-state error of the PID controller can be as high as 5.6%, but in the CTC, the controller can be reduced to 2.125%. The MSE and RMSE analysis show that CTC always responds better than PID. However, for downward movement, the response of motor 2 (M2) using CTC is better than PID.

REFERENCES

[1] G. W. Petty, R. D. Brown, J. P. Whisnant, J. D. Sicks, W. M. O'Fallon, and D. O. Wiebers, "Ischemic stroke subtypes," *Stroke*, vol. 31, no. 5, pp. 1062–1068, May 2000, doi: 10.1161/01.STR.31.5.1062.

[2] P. Rothwell *et al.*, "Population-based study of event-rate, incidence, case fatality, and mortality for all acute vascular events in all arterial territories (Oxford Vascular Study)," *The Lancet*, vol. 366, no. 9499, pp. 1773–1783, Nov. 2005, doi: 10.1016/S0140-6736(05)67702-1.

[3] T. Truelsen, B. Piechowski-Jozwiak, R. Bonita, C. Mathers, J. Bogousslavsky, and G. Boysen, "Stroke incidence and prevalence in Europe: a review of available data," *European Journal of Neurology*, vol. 13, no. 6, pp. 581–598, Jun. 2006, doi: 10.1111/j.1468-1331.2006.01138.x.

[4] D. M. Bravata, S.-Y. Ho, L. M. Brass, J. Concato, J. Scinto, and T. P. Meehan, "Long-term mortality in cerebrovascular disease," *Stroke*, vol. 34, no. 3, pp. 699–704, Mar. 2003, doi: 10.1161/01.STR.0000057578.26828.78.

[5] K. Strong, C. Mathers, and R. Bonita, "Preventing stroke: Saving lives around the world," *The Lancet Neurology*, vol. 6, no. 2, pp. 182–187, Feb. 2007, doi: 10.1016/S1474-4422(07)70031-5.

[6] P. J. O'Connor, "Trends in spinal cord injury," *Accident Analysis & Prevention*, vol. 38, no. 1, pp. 71–77, Jan. 2006, doi: 10.1016/j.aap.2005.03.025.

[7] M. P. M. van Nunen, "Recovery of walking ability using a robotic device," *Vrije Universiteit, Amsterdam*. 2013. Accessed: Jul. 20, 1BC. [Online]. Available: https://research.vu.nl/files/42114296/complete.

[8] K. H. Do and M. H. Chun, "Clinical use of robots as a part of rehabilitation medicine," *Brain & Neurorehabilitation*, vol. 10, no. 1, 2017, doi: 10.12786/bn.2017.10.e7.

[9] A. Esquenazi, "Comment on 'assessing effectiveness and costs in robot-mediated lower limbs rehabilitation: A meta-analysis and state of the art,'" *Journal of Healthcare Engineering*, vol. 2018, pp. 1–3, Nov. 2018, doi: 10.1155/2018/7634965.

[10] N. Radjou and J. Prabhu, *Frugal innovation: How to do better with less*. The Economist; First Trade Paper Edition, 2015.

[11] K. Anam and A. A. Al-Jumaily, "Active exoskeleton control systems: State of the art," *Procedia Engineering*, vol. 41, pp. 988–994, 2012, doi: 10.1016/j.proeng.2012.07.273.

[12] S. Viteckova, P. Kutilek, and M. Jirina, "Wearable lower limb robotics: A review," *Biocybernetics and Biomedical Engineering*, vol. 33, no. 2, pp. 96–105, Jan. 2013, doi: 10.1016/j.bbe.2013.03.005.

[13] M. Dzahir and S. Yamamoto, "Recent trends in lower-limb robotic rehabilitation orthosis: Control scheme and strategy for pneumatic muscle actuated gait trainers," *Robotics*, vol. 3, no. 2, pp. 120–148, Apr. 2014, doi: 10.3390/robotics3020120.

[14] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello, "Review of assistive strategies in powered lower-limb orthoses and exoskeletons," *Robotics and Autonomous Systems*, vol. 64, pp. 120–136, Feb. 2015, doi: 10.1016/j.robot.2014.09.032.

[15] B. Chen *et al.*, "Recent developments and challenges of lower extremity exoskeletons," *Journal of Orthopaedic Translation*, vol. 5, pp. 26–37, Apr. 2016, doi: 10.1016/j.jot.2015.09.007.

[16] G. Chen, C. K. Chan, Z. Guo, and H. Yu, "A review of lower extremity assistive robotic exoskeletons in rehabilitation therapy," *Critical Reviews in Biomedical Engineering*, vol. 41, no. 4–5, pp. 343–363, 2013, doi: 10.1615/CritRevBiomedEng.2014010453.

[17] M. Bernhardt, M. Frey, G. Colombo, and R. Riener, “Hybrid force-position control yields cooperative behaviour of the rehabilitation robot lokomat,” in *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, 2005, pp. 536–539, doi: 10.1109/ICORR.2005.1501159.

[18] A. K. Alshatti, “Design and control of lower limb assistive exoskeleton for hemiplegia mobility,” The University of Sheffield Mappin Street, 2019.

[19] T. Miyoshi, K. Asaishi, T. Nakamura, and M. Takagi, “Master-slave-type gait training system for hip movement disorder,” *Sensors and Materials*, vol. 28, no. 4, pp. 295–309, 2016, doi: 10.18494/SAM.2016.1180.

[20] T. P. Luu, K. H. Low, X. Qu, H. B. Lim, and K. H. Hoon, “Hardware development and locomotion control strategy for an over-ground gait trainer: NaTUre-gaits,” *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 2, pp. 1–9, 2014, doi: 10.1109/JTEHM.2014.2303807.

[21] B. Du, W. Liu, Z. Wang, and J. Chu, “A disturbance rejection controller based on maximum entropy,” in *Lecture Notes in Electrical Engineering*, 2012, pp. 155–160, doi: 10.1007/978-3-642-25778-0_23.

[22] Munadi, M. S. Nasir, M. Ariyanto, N. Iskandar, and J. D. Setiawan, “Design and simulation of PID controller for lower limb exoskeleton robot,” in *AIP Conference Proceedings*, 2018, p. 060008, doi: 10.1063/1.5046300.

[23] R. Fellag, T. Benyahia, M. Drias, M. Guiatni, and M. Hamerlain, “Sliding mode control of a 5 dofs upper limb exoskeleton robot,” in *2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)*, Oct. 2017, pp. 1–6, doi: 10.1109/ICEE-B.2017.8192098.

[24] Shiuh-Jer Huang and Ji-Shin Lee, “A stable self-organizing fuzzy controller for robotic motion control,” *IEEE Transactions on Industrial Electronics*, vol. 47, no. 2, pp. 421–428, Apr. 2000, doi: 10.1109/41.836358.

[25] Byung Kook Yoo and Woon Chul Ham, “Adaptive control of robot manipulator using fuzzy compensator,” *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 2, pp. 186–199, Apr. 2000, doi: 10.1109/91.842152.

[26] R. Jribi, B. Maalej, and N. Derbel, “Robust adaptive feedback linearization control of human exoskeletons,” in *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)*, Mar. 2019, pp. 747–751, doi: 10.1109/SSD.2019.8893163.

[27] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, doi: 10.1007/978-3-540-30301-5.

[28] D. I. Robles G., “PID control dynamics of a robotic arm manipulator with two degrees of freedom,” *Control Procesos y Robótica*, pp. 3–20, 2019.

[29] E.-H. Guechi, S. Bouzoualegh, Y. Zennir, and S. Blažič, “MPC control and LQ optimal control of a two-link robot arm: A comparative study,” *Machines*, vol. 6, no. 3, Aug. 2018, doi: 10.3390/machines6030037.

[30] A. J. McDaid, C. Lakkhananukun, and J. Park, “Paediatric robotic gait trainer for children with cerebral palsy,” in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, Aug. 2015, pp. 780–785, doi: 10.1109/ICORR.2015.7281297.

[31] T. Rahman, J. Basante, and M. Alexander, “Robotics, assistive technology, and occupational therapy management to improve upper limb function in pediatric neuromuscular diseases,” *Physical Medicine and Rehabilitation Clinics of North America*, vol. 23, no. 3, pp. 701–717, Aug. 2012, doi: 10.1016/j.pmr.2012.06.008.

[32] N. Munro and F. L. Lewis, *Robot manipulator control : Theory and practice*. 2nd ed. New York: Marcel Dekker, 2004.

[33] A. A. Martynyuk, “A survey of some classical and modern developments of stability theory,” *Nonlinear Analysis: Theory, Methods & Applications*, vol. 40, no. 1–8, pp. 483–496, Apr. 2000, doi: 10.1016/S0362-546X(00)85027-0.

[34] Z. Yang, J. Wu, J. Mei, J. Gao, and T. Huang, “Mechatronic model based computed torque control of a parallel manipulator,” *International Journal of Advanced Robotic Systems*, vol. 5, no. 1, Mar. 2008, doi: 10.5772/5650.

[35] D. Nguyen-Tuong and J. Peters, “Model learning for robot control: a survey,” *Cognitive Processing*, vol. 12, no. 4, pp. 319–340, Nov. 2011, doi: 10.1007/s10339-011-0404-1.

[36] S. J. Dodds, *Feedback control: Linear, nonlinear and robust techniques and design with industrial applications*. Springer; 1st ed., 2015.

[37] C. W. de Silva, *Modeling of dynamic systems with engineering applications*. CRC Press, 2018.

[38] M. V. M. W. Spong, *Robot dynamics and control*. Wiley, 2008.

[39] R. Paul, “Modelling, trajectory calculation and servoing of a computer controlled arm,” Stanford Artificial Intelligence Laboratory, 1972.

[40] K. M. Lynch and F. C. Park, *Modern robotics : Mechanics, planning, and control*. Cambridge: Cambridge University Press, 2017.

[41] A.-N. Sharkawy and P. Koustoumpardis, “Dynamics and computed-torque control of a 2-DOF manipulator,” *International Journal of Advanced Science and Technology*, vol. 28, no. 12, pp. 201–212, 2019.

[42] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC Press, 1994.

[43] G. Oriolo, A. De Luca, and M. Vendittelli, “WMR control via dynamic feedback linearization: design, implementation, and experimental validation,” *IEEE Transactions on Control Systems Technology*, vol. 10, no. 6, pp. 835–852, Nov. 2002, doi: 10.1109/TCST.2002.804116.

[44] Y. Meddahi, K. Zemalache Meguenni, and H. Aoued, “The nonlinear computed torque control of a quadrotor,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 20, no. 3, pp. 1221–1229, Dec. 2020, doi: 10.11591/ijeecs.v20.i3.pp1221-1229.

[45] J.-M. Belda-Lois *et al.*, “Rehabilitation of gait after stroke: a review towards a top-down approach,” *Journal of NeuroEngineering and Rehabilitation*, vol. 8, no. 1, 2011, doi: 10.1186/1743-0003-8-66.

[46] P. Sutyasadi, “Control improvement of low-cost cast aluminium robotic arm using Arduino based computed torque control,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 8, no. 4, pp. 650–659, Dec. 2022, doi: 10.26555/jiteki.v8i4.24646.

[47] Y. Meddahi and K. Z. Meguenni, “PD-computed torque control for an autonomous airship,” *IAES International Journal of Robotics and Automation (IJRA)*, vol. 8, no. 1, pp. 44–51, Mar. 2019, doi: 10.11591/ijra.v8i1.pp44-51.

[48] J. A. Shah and S. S. Rattan, “Dynamic analysis of two link robot manipulator for control design using PID computed torque control.,” *IAES International Journal of Robotics and Automation (IJRA)*, vol. 5, no. 4, pp. 277–283, Dec. 2016, doi: 10.11591/ijra.v5i4.pp277-283.

[49] A.-A. S. Abdel-Salam and I. N. Jleta, “Fuzzy logic controller design for PUMA 560 robot manipulator,” *IAES International Journal of Robotics and Automation (IJRA)*, vol. 9, no. 2, pp. 73–83, Jun. 2020, doi: 10.11591/ijra.v9i2.pp73-83.

[50] E. Jarzebowska, “Model-based tracking control of nonlinear systems,” in *Model-Based Tracking Control of Nonlinear Systems*, Chapman and Hall/CRC, 2016, pp. 205–238, doi: 10.1201/b12262-8.

[51] Z. Song, J. Yi, D. Zhao, and X. Li, “A computed torque controller for uncertain robotic manipulator systems: Fuzzy approach,” *Fuzzy Sets and Systems*, vol. 154, no. 2, pp. 208–226, Sep. 2005, doi: 10.1016/j.fss.2005.03.007.

[52] C. Chen, C. Zhang, T. Hu, H. Ni, and W. Luo, "Model-assisted extended state observer-based computed torque control for trajectory tracking of uncertain robotic manipulator systems," *International Journal of Advanced Robotic Systems*, vol. 15, no. 5, Sep. 2018, doi: 10.1177/1729881418801738.

[53] N. J. Salkind, B. B. Frey, D. M. Dougherty, K. R. Teasdale, and N. Hill-Kapturczak, *Encyclopedia of research design*. Thousand Oaks, California: SAGE Publications, 2010.

BIOGRAPHIES OF AUTHORS

Elang Parikesit received his bachelor's degree in electrical engineering from Diponegoro University, Indonesia in 1997 and a master's degree in biomedical engineering from Institut Teknologi Bandung (ITB) in 2012. He is a lecturer at the Electromedical Technology Department of Sanata Dharma University, Yogyakarta, Indonesia. He went to Rajamangala University of Technology Thanyaburi, Thailand to continue his doctoral degree in mechatronics engineering. His research interests include renewable energy, mechatronics, and biomedical engineering. He can be contacted at email: elang_p@mail.rmutt.ac.th or elang@usd.ac.id.

Dechrit Maneetham is an Associate Professor of Mechatronics Engineering at Rajamangala University of Technology Thanyaburi, Thailand. He received his B.Tech. and M.S. degrees from King Mongkut's University of Technology, North Bangkok, and his D.Eng. in Mechatronics degree from the Asian Institute of Technology in 2010, and his Ph.D. in Electrical and Computer Engineering from Mahasarakham University in 2018. His research interest includes biomedical applications, robotics and automation applications, and mechatronics applications. He can be contacted at email: dechrit_m@rmutt.ac.th.