MEKATRONIKA
e-ISSN: 2637-0883
VOL. 4, ISSUE 2, 6 – 15
DOI: https://doi.org/10.15282/mekatronika.v4i2.8941

ORIGINAL ARTICLE

Design A Precision Motion Control of an Upper Limb Robotic Arm

Kwan Ze Zhen[1], Ismail Bin Mohd Khairuddin[1,*], Mohd Azraai Mohd Razman[1], Anwar P.P. Abdul Majeed[1], and Wan Hasbullah Mohd Isa[1]

[1]Faculty of Manufacturing Engineering, Universiti Malaysia Pahang, 26600 Pahang, Malaysia.

ABSTRACT – Trajectory tracking is utilized in medical rehabilitation programs at the early stage of rehabilitation in order to track the performance of the patient in performing the prescribed task. The robotic arm has been utilized to accomplish this due to it precision and provide repetitive motion. The goal of this study is to design and simulate a two-degree-of-freedom robotic arm that can effectively track a trajectory. As a result, this study discusses the modelling, simulation, and control of a Two Degree of Freedom (2-DOF) Robot Arm to attain that goal. First, the robot specifications are provided, as well as the forward and inverse kinematics of a 2-DOF robot arm. The dynamics of the 2-DOF robot arm were then defined using the Euler- Lagrange Equation to obtain motion equations. A PID controller was used to construct a control design for the robot's controller. MATLAB is used to record all the data, including the margin of error, overshoot, and peak settling time. The data is identified using the PI and PID controllers, in which the error is smaller than 7 and 1.5, respectively. The controller was then used to create a prototype model by using MATLAB Sim Mechanics. The data obtained indicates that a PID controller is the best fit for this rehabilitation robot.

ARTICLE HISTORY
Received: 12th Aug 2022
Revised: 4th Sept 2022
Accepted: 4th Oct 2022

KEYWORDS
Upper limb
Robotic Arm
Controller
Motion Control

INTRODUCTION

Robotic arm is essential technology nowadays and has been widely used in many areas of industry. This is because the robotic arm can be any of several mechanical, programmable devices and others. The robotic arms, also known as articulated robotic arms, are quick, dependable, and precise machines that can be taught to do several tasks in various conditions. A robotics arm is frequently made up of different degrees of freedom, which refers to several moveable joints of a robot. The number of distinct displacements or characteristics of motion necessary to fully define a robotics arm's configuration is the same as the number of degrees of freedom. For example, the Cartesian coordinates of sites, the joints angles connections, or a mix of both Cartesian coordinates and joints angles can be used to describe a mechanism.

Furthermore, the robotic arm required motion control to complete one of the most critical phases: doing the activity. Aside from that, motion control is a sub-field of automation that encompasses the systems or sub-systems engaged in the controlled movement of machine parts, generally with highly exact speeds, locations, and torque control. To do a task that has been established, for example, the action of the rotating joints, motion control is one of the steps for a robotic system that specifies how a robotic arm should perform the task.

To make a robotic arm helpful in life, the mechanical characteristics of the robotic arm should be exploding. This is because the mechanical properties of a robotics arm show how the object is performing a good performance under different conditions or upon application of load and force or the movement of the object that will help to identify the suitability for different applications.

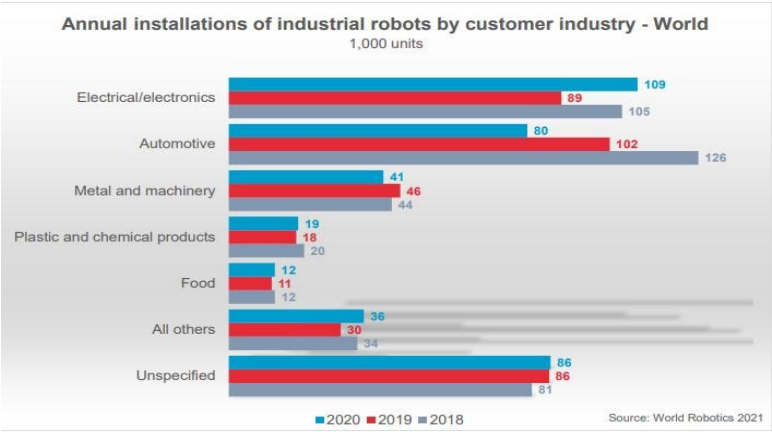

Figure 1. Statistic of Robotic Arm

*CORRESPONDING AUTHOR | Ismail Mohd Khairuddin | ismailkhai@ump.edu.my

According to the World Robotics 2021 Industrial Robots report, shown in Figure 1, there are around three million industrial robots in use in factories worldwide, representing a 10% growth. The automobile industry, for example, has the most installed robotics units in the world, accounting for roughly 42% of all robots.

The International Federation of Robotics (IFR) defines a robotic arm application as a robotic system designed for manufacturing applications that can be automated and programmed to do tasks without a human controller. Industrial robotic arms are commonly used for welding, assembling, picking, placing, packing, labelling, palletizing, and product assembly. The industrial and automotive industries are the most prevalent places to see robotic arms.

An upper limb robotics arm's accurate motion control depends on the mechanism's mechanical features, degrees of freedom, type of actuators and motors, sensor, and controller. The two types of robotic arms are exoskeleton-type and end-effector type. Exoskeleton-type can map motion and torque to the corresponding human joint, allowing the robotics arm to provide better direction and control the individual joints. In contrast, end effector-type are unable to map onto the corresponding human joints, limiting the robotics arm's ability to produce whole arm motion[6] . Furthermore, compared to the end-effector and the exoskeleton, the exoskeleton has a more excellent range of motion (ROM), allowing for more realistic movement in everyday situations [2] .

Aside from that, the kind of exoskeleton actuation and control is a critical component in supplying a portable feature. Actuators can deliver the needed torque to specific joints, allowing individuals to move their limbs more freely. The controller affects the motion of the robotic arm, precision, time setting, steady-state error, and much more in the motion control of an upper limb's robotics arm. When the robotics arm is programmed, the controller determines how and in which direction the arm should move. Electric, hydraulic, and pneumatic actuators, as well as controllers such as fuzzy logic, PID, and others, are tested for their impact on the mechanical characteristics of the robotics arm.

This study focuses on the motion control of an upper limb robotic arm. Besides that, the performance of the controller of the robotic arm will be discussed briefly in the next chapter.

RELATED WORK

There are two types of robotic arms: exoskeleton devices shown in Figure 2 and end-effector shown in Figure 3. Exoskeleton of the robotic arm have a vector whose components are the translational and angular displacement of each joint of a mechanical connection that defines joint space while end-effector define Cartesian space via its location and orientation, such as a function that is written using (x, y) or (x, y, z) coordinates.

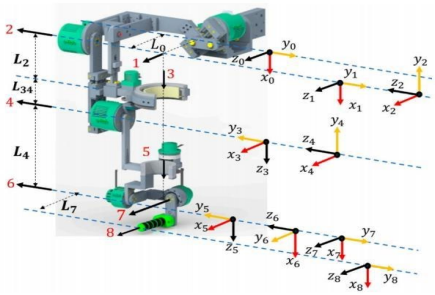

Figure 2. Exoskeleton Robotic Arm

Figure 3. End-Effector Robotic Arm

Exoskeletons in robotics are divided into two categories: upper limb exoskeletons and lower limb exoskeletons. The motion and torque of the corresponding human joint can be mapped by exoskeleton devices in the upper limb extremities, leading in enhanced guiding and control of the individual joints. End-effector devices, such as grippers, which are the most common type, however, are unable to map onto human joints because they are unable to perform a full arm motion. It would be a good end-point exercise and easy to adapt to spastic people with rotated shoulders, flexed elbows, pronated forearms, and wrists [4].

The amount of degrees of freedom determines the robotic arm's complexity (DOF). The exoskeleton and end-effector devices will be limited in their ability to provide an extensive range of movement, which will be dependent on degrees of the robotic arm's upper limb. This implies that the multiple degrees of freedom of the robotic arm, such as the joint, are used to control the different components of the robotic arm's movement. Resolution of the extra DOF is needed, and performance index enhancement need to be implemented; according to [5], it is difficult to control redundant movements of upper extremity exoskeletons better.

Furthermore, according to [4] end-effector robots drive the upper limb for rehabilitation training activities by contacting the upper limb extremity with it, while exoskeleton robots regulate the exact movement of the patient's arm in several joints with a more complex mechanical construction. However, according to [9], end-effector rehabilitation robots have just one link between the patient's hand and the end-effector. Therefore, it is unable to discern the damaged limb's configurations independently. To verify that an exoskeleton-type rehabilitation robot's mechanical construction is more sophisticated, as it imitates the human skeleton and ensures that the human and robot joint axes are aligned. Exoskeletons with many contact points can be worn on the afflicted limb, allowing the assisting torque operating on each human joint to be adjusted independently.

However, the interaction force between robot ends and affected limb is critical in the rehabilitation robot control systems design. Given the extremely nonlinear nature of human-robot interaction movement in the rehabilitation field, achieving consistently good tracking performance in passive training with disturbances and uncertainties is a significant difficulty. To avoid the difficulty [5], exoskeletons must avoid large-scale variations in potential energy during the motion control. Based on the research by [3],[10] the exoskeletons are a common form of exoskeleton that has been extensively researched in the literature. They're mostly utilised to treat upper-limb sports injuries or people who have had strokes that have left them with upper-limb disabilities. Due to this the exoskeleton robotic arm is the choice to use to design a motion control of the upper limb robotic arm. Exoskeleton acts as a human- like anatomy used to rehabilitate and facilitate human upper limb mobility in rehabilitation robots. Therefore, the use of exoskeleton to interact depends on the quality of the accompanying controllers to obtain satisfactory results.

Controller Implementation of Robotic Arm

There are many implementations of a controller in the world, such as PID Controller, Fuzzy logic Controller, Sliding mode controller and others. A robotic arm requires accurate motion control to establish the trajectory and torque required to produce a certain output. PID controller calculates the difference between planned set points and actual output and the process is then corrected due to the calculation. The control law of P, I, D is shown as below in Table 1.

Table 1. Control Law of PID Control

Controller	Mathematical equation
PI Controller	$U(t) = K_{p2}T_i(t) + K_{i2}\int T_i(t)dt$
PD Controller	$U(t) = K_{p1}T_i(t) + K_{d1}\frac{dT_e(t)}{dt}$
PID Controller	$(t) = K_p e(t) + \frac{1}{T_I}\int_0^1 e(t)dt + T\frac{d_e(t)}{dt}$

The present upper limb robots, according to [4] can quantify upper arm forces and monitor interaction forces during passive rehabilitation exercises are both lacking. This research aims to develop an upper arm cuff that can force the wearer's upper arm. A PID control method was employed for both joint-based and end-point exercises. The tracking inaccuracy for both joint-based workouts and end-point activities was astonishingly low after validating the exoskeleton and end-effector aspects of the produced prototype.

In the other hand, in [1] proposed that using trapezoid fuzzy PID algorithm to control the robotic arm. The trapezoid fuzzy PID method is based on the error rate adjustment P, I, D value model based on symmetry principles for control of robot arm motion with fault diagnostics. The proposed controller has made PID control improve the tracking system's stability by allowing logical reasoning and computation. PID control is a proportional, integral, differential, and linear controller. In this method, it can notice that the connection of each parameter and the rate of error change and the system error.

The smoothness control of the robotic arm trajectory still has certain flaws due to the PID controller's limitations. The PID controller may be used for a range of control activities, but it does not perform well when it comes to optimum control. According to previous study, an adaptive controller merging fuzzy set methods and a PD controller is employed to enhance trajectory tracking [7]. The traditional PD controller is often used in exoskeleton control since it is model agnostic, and gains can be easily changed.

However, in comparison, based on research [8] , DC geared motor was used in the design and fabrication of one degree of freedom robotic arm. In the prototype, the PID controller and fuzzy controller are compared based on their ability in terms of steady-state error, settling time, rise time and overshoot percentage in the point to point (PTP) control performances. In the experiment, the PID controller is investigated based on the Ziegler-Nicholas frequency response and it will evaluate the robot manipulator's capabilities and precisely control its motion. In several types of experiments such as open-loop systems, uncompensated systems, and compensated systems, the PID controller is better at removing steady-state errors, but the fuzzy logic controller has a faster setting time than the PID controller. The rise time (Ts) of the Fuzzy Logic controller is longer than that of the PID controller. The PID controller achieves good performance whereas the steady-state error is less than 0.01. The setting time of the PID Controller is 0.5s and for the input reference is about 150.

METHODOLOGY

The proposed framework for this research is focused on the process of how to design a controller for an upper limb robotic arm to achieve a minimum error. This research flow is divided into three phases. Phase one is to define the problem and identify the literature review on the robotic arm's type of controller. Phase two involves the formulation of the mathematical equation of two degrees of freedom of the robotic arm, followed by defining the type of technique used to calculate the dynamic equation. Next, the final phases are the data will be simulated using MATLAB Simulink by four

types of cases: without a controller, with a controller, simulate the CAD drawing without a controller and with a controller. The final step is interpretation and conclusion. The flow chart of this research is then shown in Figure 4.

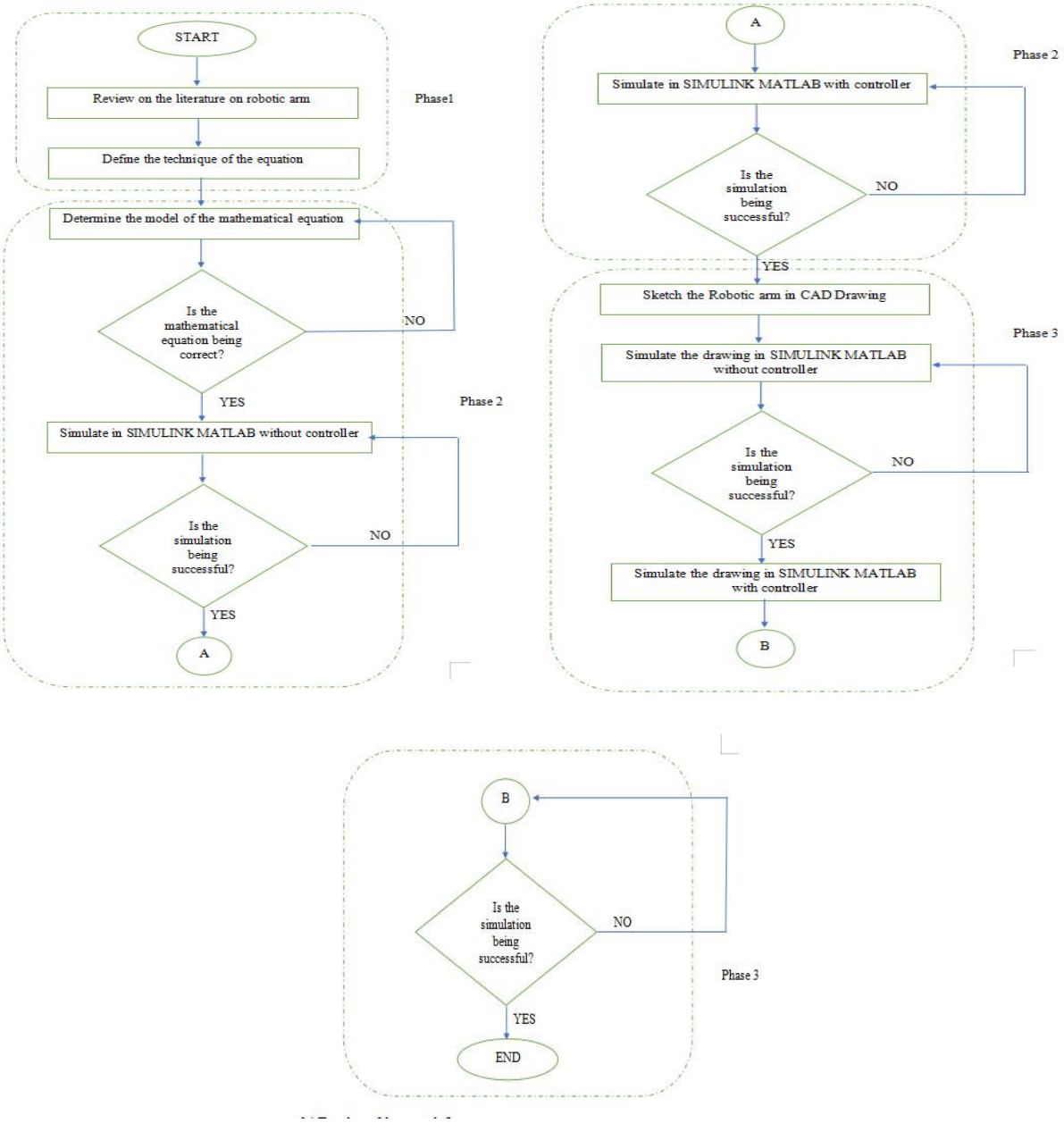

Figure 4. Flow Chart of Research

Modelling of the Mathematical Equation

In this section, the mathematical equation of the two degrees of freedom is formulated using the technique based on the Euler-Lagrange equation shown in Figure 5.

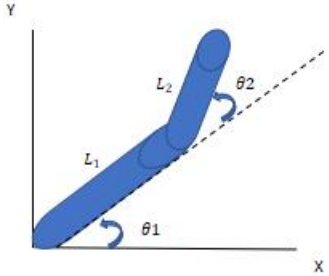

Figure 5. Coordinate of 2 Link Robotic Arm

Coordinate frame as:

$$x = L_1 cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) \quad (1)$$
$$y = L_1 sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) \quad (2)$$

in which L_1 and L_2 are the length of two links, respectively where θ_1 and θ_2 is the angle of the robotic arm. Next, integral the equation 1 and 2:

$$\dot{x} = -L_1 sin\dot{\theta}_1(\dot{\theta}_1) - L_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \quad (3)$$
$$\dot{y} = L_1 sin\dot{\theta}_1(\dot{\theta}_1) + L_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \quad (4)$$

Summation of the vector magnitude for velocity:

$$V^2 = \dot{x}^2 + \dot{y}^2 \quad (5)$$

From the equation 5:

$$\dot{x}^2 = (-L_1 sin\dot{\theta}_1(\dot{\theta}_1) - L_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2))^2 \quad (6)$$

$$\dot{y}^2 = (L_1 cos\dot{\theta}_1(\dot{\theta}_1) + L_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2))^2 \quad (7)$$

Then, add the $\dot{x}^2$ and $\dot{y}^2$

$$V^2 = L_1{}^2\dot{\theta}_1{}^2 + 2L_1L_2 cos\,\theta_2 \cos\left(\left(\dot{\theta}_1{}^2 + \dot{\theta}_1\dot{\theta}_2\right) + L_2{}^2\right) + \left(\dot{\theta}_1{}^2 + 2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2{}^2\right) \quad (8)$$

Hence the kinetic energy is given will be:

$$K = \frac{1}{2}mv^2 \quad (9)$$

From the equation nine the kinetic energy of the first and second link will be:

$$K_1 = \frac{1}{2} m_1 L_1{}^2\dot{\theta}_1{}^2 \quad (10)$$

Then,

$$K_2 = \frac{1}{2} m_1 L_1{}^2\dot{\theta}_1{}^2 + \frac{1}{2} m_2 L_2{}^2\left(\dot{\theta}_1{}^2 + 2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2{}^2\right) + m_2 L_1 L_2 cos\,\theta_2\left(\dot{\theta}_1{}^2 + \dot{\theta}_1\dot{\theta}_2\right) \quad (11)$$

Total Kinetic energy:

$$K = K_1 + K_2 \quad (12)$$

$$K_T = \frac{1}{2} m_1 L_1{}^2\dot{\theta}_1{}^2 + \frac{1}{2} m_1 L_1{}^2\dot{\theta}_1{}^2 + \frac{1}{2} m_2 L_2{}^2\left(\dot{\theta}_1{}^2 + 2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2{}^2\right) + m_2 L_1 L_2 cos\,\theta_2\left(\dot{\theta}_1{}^2 + \dot{\theta}_1\dot{\theta}_2\right) \quad (13)$$

The manipulator's potential energy is thus equal to the total of two linkages' potential energies. The potential energy of each link is equal to its mass multiplied by gravitational acceleration and the height of its mass. Thus

$$P_1 = m_1 g L_1 sin\theta_1 \quad (14)$$
$$P_2 = m_2 g(L_1 sin\theta_1 + L_2 sin(\theta_1 + \theta_2)) \quad (15)$$
$$P = m_1 g L_1 sin\theta_1 + m_2 g(L_1 sin\theta_1 + L_2 sin(\theta_1 + \theta_2)) = (m_1 + m_2) g L_1 sin\theta_1 + m_2 g L_2 \sin(\theta_1 + \theta_2)) \quad (16)$$

Then the Euler-Lagrange for the system will be:

$$L = \tfrac{1}{2}(m_1 + m_2) L_1{}^2\dot{\theta}_1{}^2 + \tfrac{1}{2} m_1 L_1{}^2\dot{\theta}_1{}^2 + \tfrac{1}{2} m_2 L_2{}^2\left(\dot{\theta}_1{}^2 + 2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2{}^2\right) + m_2 L_1 L_2 cos\,\theta_2\left(\dot{\theta}_1{}^2 + \dot{\theta}_1\dot{\theta}_2\right) - (m_1 + m_2) g L_1 sin\theta_1 + m_2 g L_2 \sin(\theta_1 + \theta_2)) \quad (17)$$

After that, differentiate the Lagrange of the equation 17 for the first link of the robotic arm:

$$\frac{dL}{d\dot{\theta}_1} = (m_1 + m_2) L_1{}^2\dot{\theta}_1 + m_2 L_2{}^2(\dot{\theta}_1 + \dot{\theta}_2) + 2m_2 L_1 L_2 cos\,\theta_2(\dot{\theta}_1) + m_2 L_1 L_2 cos\,\theta_2(\dot{\theta}_2) \quad (18)$$

Then differentiate again in second order of the θ .

$$\frac{dL}{d\dot{\theta}_1} = (m_1 + m_2) L_1{}^2\dot{\theta}_1 + m_2 L_2{}^2\dot{\theta}_1 + m_2 L_2{}^2\dot{\theta}_2 + 2m_2 L_1 L_2 cos\,\theta_2(\dot{\theta}_1) + m_2 L_1 L_2 cos\,\theta_2(\dot{\theta}_2) - 2m_2 L_1 L_2 sin\,\dot{\theta}_1(\dot{\theta}_1) - m_2 L_1 L_2 sin\,\dot{\theta}_2(\dot{\theta}_2)$$
$$= \left[(m_1 + m_2) L_1{}^2 + m_2 L_2{}^2 + 2m_2 L_1 L_2 cos\theta_2\right]\ddot{\theta}_1 + \left[m_2 L_2{}^2 + m_2 L_1 L_2 cos\theta_2\right]\ddot{\theta}_2 - 2m_2 L_1 L_2 sin\theta_2\dot{\theta}_1{}^2 - m_2 L_1 L_2 sin\theta_2\dot{\theta}_2{}^2 \quad (19)$$

Differentiate the gravitational energy based on equation 16:

$$\frac{dL}{d\dot{\theta}_1} = (m_1 + m_2) g L_1 cos\theta_1 + m_2 g L_2(\theta_1 + \theta_2) \quad (20)$$

From the above, the first equation of the first link motion will be:

$$T_1 = \left[(m_1 + m_2)\, {L_1}^2 + m_2 {L_2}^2 + 2m_2 L_1 L_2 cos\theta_2\right]\ddot{\theta}_1 + \left[m_2 {L_2}^2 + m_2 L_1 L_2 cos\theta_2\right]\ddot{\theta}_2 - 2m_2 L_1 L_2 sin\theta_2 \dot{\theta}_1^{\,2} - m_2 L_1 L_2 sin\theta_2 \dot{\theta}_2^{\,2} + (m_1 + m_2) g L_1 cos\theta_1 + m_2 g L_2 (\theta_1 + \theta_2) \quad (21)$$

For the second link of the robotic arm,

$$\frac{dL}{d\dot{\theta}_2} = m_2 {L_2}^2 (\theta_1 + \theta_2) + m_2 L_1 L_2 cos\,\theta_2\, \dot{\theta}_1 \quad (22)$$

Differentiate into second order of θ

$$\frac{dL}{d\dot{\theta}_2} = m_2 {L_2}^2 \left(\ddot{\theta}_1 + \ddot{\theta}_2\right) + m_2 L_1 L_2 cos\,\theta_2\, \dot{\theta}_1 - m_2 L_1 L_2 sin\theta_2 \dot{\theta}_1^{\,2} \quad (23)$$

Differentiate the gravitational energy based on equation 16:

$$\frac{dL}{d\dot{\theta}_2} = m_2 L_1 L_2 cos\theta_2 \left(\dot{\theta}_1^{\,2} + \dot{\theta}_1 \dot{\theta}_2\right) + m_2 g L_2 \cos(\theta_1 + \theta_2) \quad (24)$$

From the above, the equation of 2nd link of the robotic arm will be,

$$T_2 = \left[m_2 {L_2}^2 + m_2 L_1 L_2 cos\theta_2\right]\ddot{\theta}_1 + m_2 {L_2}^2 \ddot{\theta}_2 + m_2 L_1 L_2 sin\theta_2 \dot{\theta}_1^{\,2} + m_2 g L_2 \cos(\theta_1 + \theta_2) \quad (25)$$

CAD Drawing

This section explains the CAD drawing for the two degree of freedom robotic arm which will perform shoulder and elbow movement. The robotic arm's design has been drawn using the Autodesk Inventor. The concept of the robotic arm is to help people manage the hand's movement or recover the hand's motion. To begin, the mechanical system of the robotic arm was developed to closely resemble of the ideal system, which is that the arm is very light in weight, has an effective point mass at its tip, and has low friction and backlash. In the design, the dimension of the parts of the robotic arm is considered according to a human arm and it features an ergonomic design that allows the length of the mechanism to be adjusted to the user's arm length. In addition, the robotic arm will implement two controllers to control the robotic arm, such as the direction of the robotic arm and the angle of the robotic arm that can apply in the system. Having the controller will support and control the movement and direction of the hand to achieve the objective or the target.

For a complete prototype, all parts will be designed and then assembled. These components must be carefully designed for the assembly to be simple. A complete prototype is shown in the following Figure 6.

Figure 6. Overview of Robotic Arm

Simulation

The experiment is carried out using MATLAB SIMULINK simulation for the simulation. Create a block diagram for each matrix in the dynamic equation to begin. These block diagrams will be used in the upper limb system, which is also known as the plant. These are the variables that were utilized. The block diagram of the simulation will be shown in Figure 7.

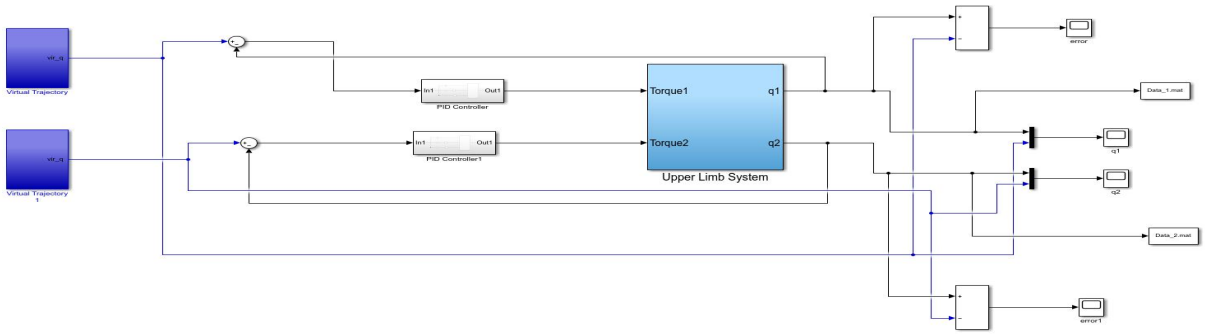

Figure 7. Simulation Overall Design

Table 2. Parameter for Mass and Length for Link 1 and Link 2

Performance	Link 1	Link 2
Link 1, *m*	1.91	1.22
Link 2, *m*	0.34	0.25
Mass, *kg*	9.81	9.81

Performance

The performance of the robotic arm will be analysed in terms of the Root mean error (RMS), Settling time (Ts), Rise Time (Tr), and the Percentage of the overshoot (%OS). The formula is shown as below:

$$RMS = \sqrt{\sum_{i=1}^{n} \frac{(\check{y}_i - y_i)^2}{n}} \qquad Tr = \frac{1.8}{W_n} \qquad Ts = \frac{4}{\sigma} = \frac{4}{\zeta W_n} OS = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \tag{26}$$

; where $\check{y}_i$ = predicted value y_i = observed value n = Number of observations, W_n = Natural Frequency

Experiment Results

This section will show the PID Controller's result, which controls the upper limb robotic arm. It will show the PID Controller's accuracy which follows the trajectory tracking, which is set by the Input. In the Figure 8 it will show the parameter of the PID Controller. In the simulation, it will be used to control the motion of the robotic arm.

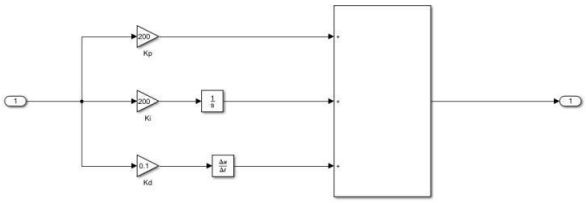

Figure 8. PID Controller in MATLAB SIMULINK

This section will consist of simulation results of the robotic arm such as one motion and repeated motion of the PID Controller which to show that the accuracy of the controller to track the motion and to compare with the simulation without controller.

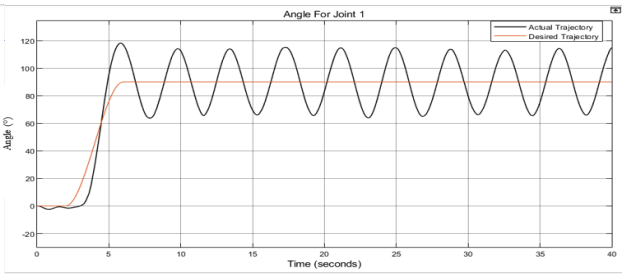

Figure 9. Simulation of Joint 1 Without Controller

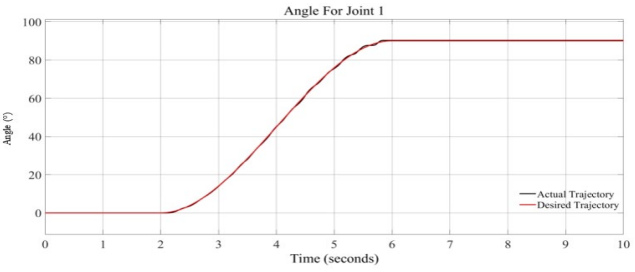

Figure 10. Simulation Joint 1 With PID Controller

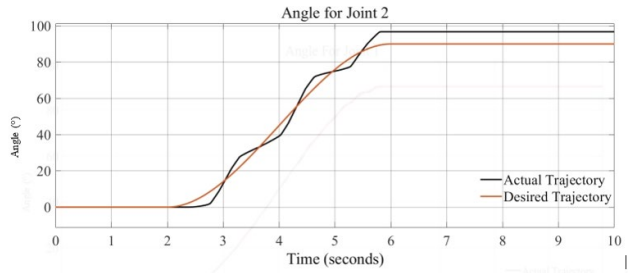

Figure 11. Simulation of Joint 2 Without Controller

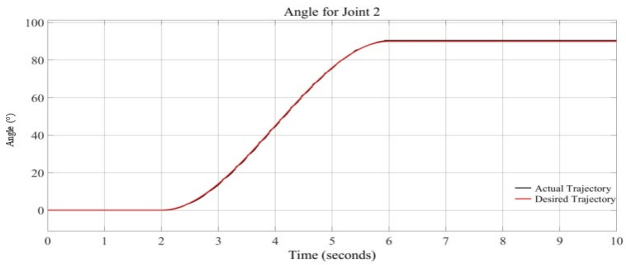

Figure 12. Simulation of Joint 2 With PID Controller

Table 3. Performance of Robotic Arm (One motion)

Performance	Joint 1	Joint 2
Rise Time, T_R	0.3712 s	0.3392 s
Overshoot, *OS%*	-	-
Peak Time, T_P	6.02 s	6.41 s
Root Mean Square, *RMS*	1.334e-01	2.893e-01

Repeated Motion of robotic arm using PID Controller

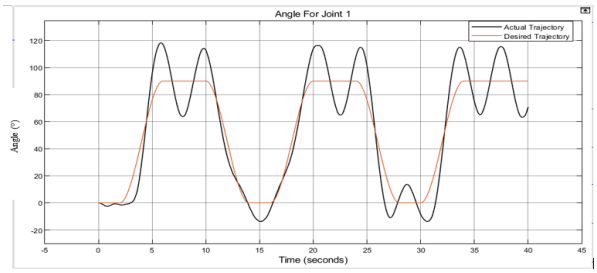

Figure 13. Simulation of Joint 1 without Controller

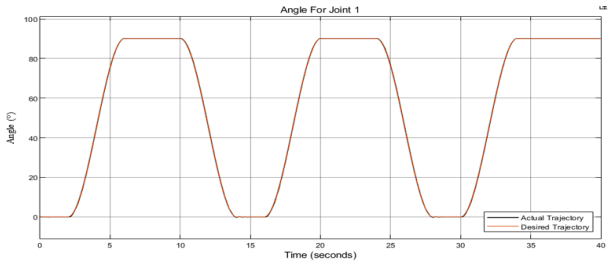

Figure 14. Simulation of Joint 1 with PID Controller

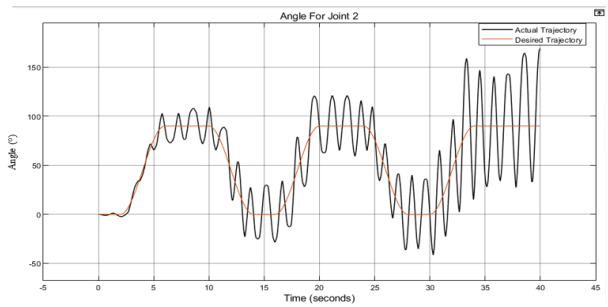

Figure 15. Simulation of Joint 2 Without Controller

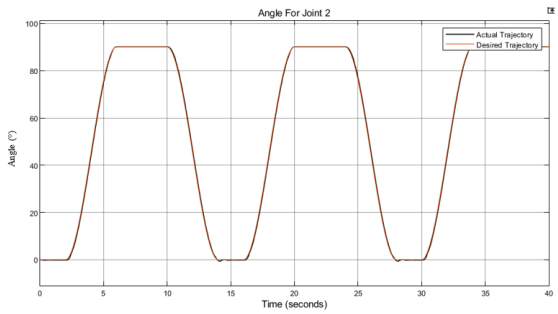

Figure 16.: Simulation of Joint 2 with PID Controller

Table 4. Performance of Robotic Arm (Repeated Motion)

Performance	Joint 1	Joint 2
Rise Time, T_R	0.3712 s	0.3392 s
Overshoot, *OS%*	-	-
Peak Time, T_P	6.02 s	34.41 s
Root Mean Square, *RMS*	2.928e-01	6.287e-01

In the Figures 9 and 11 show that simulation result of one motion without PID controller and Figure 13 and 15 is the simulation of repeated motion without PID controller whereas the Figure 10 and 12 is the simulation result with PID controller and Figure 14 and 16 is the simulation of repeated motion with PID Controller. Based on the simulation result for one motion and repeated motion of PID Controller, the controller perfectly controls the robotic arm to get the better trajectory. From the Figure 10 and Figure 12 show that the trajectory remains at zero when it is in 2 seconds before it starts. While the trajectory increases constantly in the time from 2 seconds until 6 seconds, it remains in 90 degree after 6 seconds and will start to decrease to 0 degree during the time 10s until 14s. It will repeat the same motion until 40s for

a repeated motion of controller. Without any type of disturbance, the joint root means square error (RMS) for the system governed by the PID controller is 2.928 x 10^{-1}at joint 1 and 6.287 x 10^{-1}at joint 2 for repeated motion, whereas for one motion is 3.395x 10^{-1}at joint 1 and 3.774 x 10^{-1}at joint 2. The rise time for repeated motion the joint 1 and joint 2 is 0.3712 and 0.3392 respectively, shown in Table 3 whereas for one motion, the rise time is 0.3741 and 0.4071 respectively, shown in table 4. The findings show that the PID control method for the joint one the peak time is about 6.02s, while the 34.41s for the joint 2. In addition, the percentage overshoot of the repeated motion and one motion for the joint 1 and joint 2 has a small probability which is 1.4012 x 10^3 and 1.4014 x 10^3respectively in repeated motion while for one motion is 1.4071 x 10^3 and 1.4084 x 10^3 respectively. The results reveal that the proposed control system can properly adjust any form of disturbance while maintaining good joint tracking error.

Sim Mechanics Simulation

The initial position is located vertically from the body. With the cycle, the arm is configured to travel 90 degrees outside the body. The first rotation will be slower than the second to see if the quicker motion will alter the controller's movements and lastly, it will return to its original position shown in the Figure 17 and Figure 18. This action is repeated with a different type of controller for each signal.

Figure 17. Initial Position

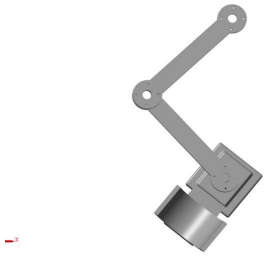

Figure 18. Final Position

Simulation Result Using Sim Mechanics

This section will consist of the simulation of Sim Mechanics for the robotic arm's Two link motion and One link motion. The simulation results show that, the two-link motion of the PID controller have a better accuracy to follow the trajectory motion compared with the Input that have been calculated. During the Sim Mechanics simulation, the trajectory increases accurately in the time the simulation starts. The robotic arm will be configured to travel 90 degrees outside the body and return to its original position at final. The findings of the PID Controller will be shown in the Table 5.

Table 5. Performance of robotic arm with PID Controller (Two link Motion)

Performance	Joint 1	Joint 2
Rise Time, T_R	0.9628 s	2.0604 s
Overshoot, *OS%*	-	-
Peak Time, T_P	8.96 s	21.60 s
Root Mean Square, *RMS*	6.180e+01	6.180e+01

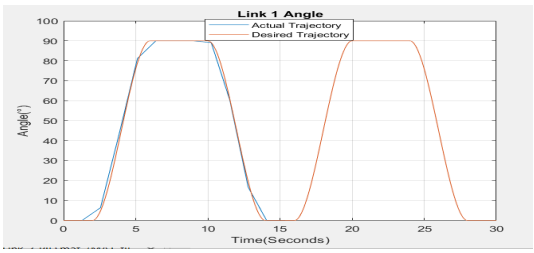

Figure 19. Comparison of Input with PID Controller (Link 1-two link motion)

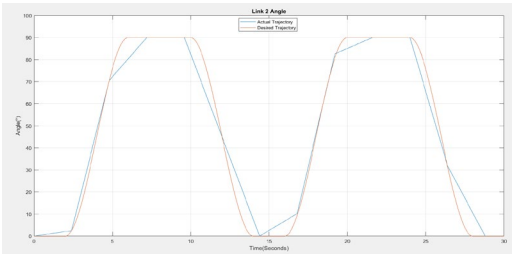

Figure 20. Comparison of Input with PID Controller (Link 2 -two link motion)

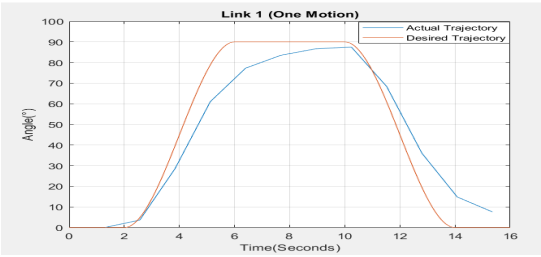

Figure 21. Comparison of Input with PID Controller (one link motion)

As of one link motion of robotic arm with the PID Controller has a good trajectory motion control and a similar good accuracy compared with the Input that calculated. While the trajectory increases in a similar good accuracy in the time the simulation starts. The robotic arm will be configured to travel 90 degrees to the outside of the body, and it will return to its original position at final while the finding of the control methods will be shown in the Table 6.

Table 6. Performance of one link motion of Robotic arm with PID Controller

Performance	Joint 1
Rise Time, T_R	1.1592 s
Overshoot, *OS%*	-
Peak Time, T_P	10.24 s
Root Mean Square, *RMS*	5.765e+01

CONCLUSION

As shown by this study goals, the overall project objectives have been accomplished. The first part of this study explains how a linearized mathematical model generated from kinematic and dynamic equations was used to develop a 2-DOF robotic manipulator. The Euler-Lagrange equations were used to create a dynamic model that accurately reproduced real-world robot movement while also giving the robot joint positions enough motion control. To regulate the system and obtain the desired joint angle position, PID controller simulations in MATLAB/SIMULINK were applied. Once the criteria are fulfilled, the data manipulation will start. The research authors revealed that the proposed 2- DOF robotic manipulator is particularly successful when paired with PID feedback. In addition, several controller types were investigated to further develop the 2-DOF robotic manipulator. A more complex design simulation might be employed to prototype the project. These virtual prototypes were constructed with Inventor and then analysed with MATLAB Sims Mechanic for a more in-depth investigation. These have shown to be the most effective methods for prototyping concepts. In the result show that PID achieved a good result in terms of feedback, with a little projectile motion error. Lastly, PID controller gains must be updated and managed according to their use to minimize overshoot and oscillation caused by changes in parameter values.

REFERENCES

[1] Bi, M. (2020). Control of robot arm motion using trapezoid fuzzy two-degree-of-freedom PID algorithm. *Symmetry*, *12*(4). https://doi.org/10.3390/SYM12040665

[2] Chen, S. H., Lien, W. M., Wang, W. W., Lee, G. de, Hsu, L. C., Lee, K. W., Lin, S. Y., Lin, C. H., Fu, L. C., Lai, J. S., Luh, J. J., & Chen, W. S. (2016). Assistive control system for upper limb rehabilitation robot. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *24*(11), 1199–1209. https://doi.org/10.1109/TNSRE.2016.2532478

[3] Fallaha, C., Saad, M., Ghommam, J., & Kali, Y. (2021). Sliding Mode Control with ModelBased Switching Functions Applied on a 7-DOF Exoskeleton Arm. *IEEE/ASME* Transactions *on Mechatronics*, *26*(1), 539–550. https://doi.org/10.1109/TMECH.2020.3040371

[4] Islam, M. R., Assad-Uz-Zaman, M., Brahmi, B., Bouteraa, Y., Wang, I., & Rahman, M. H. (2021). Design and Development of an Upper Limb Rehabilitative Robot with Dual Functionality. Micromachines, 12(8), 870. https://doi.org/10.3390/mi12080870

[5] Li, Z., Zuo, W., & Li, S. (2020). Zeroing dynamics method for motion control of industrial upper-limb exoskeleton system with minimal potential energy modulation. *Measurement: Journal of the International Measurement Confederation*, *163*. https://doi.org/10.1016/j.measurement.2020.107964

[6] Md Ghazaly, M., Teo, T. H., A/L Regeev, V., A/L Vijayan, K., Shin Hong, C., Che Amran, A., Abdullah, Z., & Md Ali, M. A. (2016). POINT-TO-POINT (PTP) CONTROL PERFORMANCES OF AN UPPER LIMB ROBOTIC
ARM. Jurnal Teknologi, 78(7). https://doi.org/10.11113/jt.v78.7172.

[7] Tang, J., Zheng, J., & Wang, Y. (2018). Direct force control of upper-limb exoskeleton based on fuzzy adaptive algorithm. *Journal of Vibroengineering*, *20*(1), 636–650. https://doi.org/10.21595/jve.2017.18610

[8] Teknologi, J., Md Ghazaly, M., Ting Huan, T., Regeev, V. A. / L., Vijayan, K. A. / L., Horng, C. S., Amran, A. C., Abdullah, Z., Amran, M., & Ali, M. (2016). *POINT-TO-POINT (PTP) CONTROL PERFORMANCES OF AN UPPER LIMB ROBOTIC ARM* (Vol. 78, Issue 7). www.jurnalteknologi.utm.my

[9] Wu, Q., Wang, X., Chen, B., & Wu, H. (2018). Development of an RBFN-based neural-fuzzy adaptive control strategy for an upper limb rehabilitation exoskeleton. *Mechatronics*, *53*, 85–94. https://doi.org/10.1016/j.mechatronics.2018.05.014

[10] Mohd Fauzi, M. H., Mohd Khairuddin, I., P. P. Abdul Majeed, A., Mohd Razman, M. A., & Mohd Isa, W. H. (2021). Articulated Robot Arm. MEKATRONIKA, 3(2), 57–64. https://doi.org/10.15282/mekatronika.v3i2.7354