International Journal of Membrane Science and Technology, 2023, Vol. 10, No. 3, pp 2448-2459

Robotic Arm Design and Control Using MATLAB/Simulink

Rawiphon Chotikunnan[1], Kittipan Roongprasert[2*], Phichitphon Chotikunnan[3], Pariwat Imura[4], Manas Sangworasil[5], Anuchart Srisiriwat[6]

[1,2,3,4,5]*College of Biomedical Engineering, Rangsit University, Pathum Thani, Thailand 12000; Email: rawiphon.c@rsu.ac.th*

[6]*Department of Electrical Engineering, Pathumwan Institute of Technology, Bangkok, Thailand 10330*

Abstracts: This research focuses on leveraging the capabilities of MATLAB/Simulink and Arduino microcontrollers to develop a control system for a robotic arm intended for medical and industrial applications. The arm's structural framework consists of three motors, each connected to adjustable resistors to form a comprehensive servo motor system. By integrating these hardware components with software solutions, the study aims to create a flexible, precise, and reliable automation system. The system's position and rotational control are executed through an Arduino microcontroller, which communicates with a computer running MATLAB/Simulink software. This configuration allows for real-time data processing and system adjustments. One of the study's key contributions is the utilization of Trajectory Control techniques, which govern the arm's movements through pre-defined paths, ensuring optimal efficiency and accuracy. Furthermore, the study introduces the use of a Smoothing Function to mitigate system over-shoot, thereby enhancing control precision. The research validates its methodologies through a series of tests. Results indicate that the robotic arm successfully navigates to predetermined positions with error magnitudes as low as 2.8587, 5.7340, and 4.4406 in the A, B, and C motor axes, respectively. These outcomes demonstrate the system's potential for high-precision tasks in medical and industrial settings.

Keywords: MATLAB/Simulink, Robotic Arms, Trajectory Control.

1. INTRODUCTION

In contemporary times, robotics has become an integral part of human life, penetrating various sectors from healthcare to industrial applications. Rapid advancements in robotic technology have enabled robots to perform tasks that are hazardous or beyond human capabilities, such as disaster relief, bomb disposal, and chemical processing. One crucial aspect of robotic development is motion control, which encompasses hardware and software systems that drive the robot's movements. Software such as MATLAB/Simulink [1]-[4] offers comprehensive capabilities ranging from algorithm development and mathematical modeling to system simulation. It is widely applied in electrical engineering, energy storage, and complex system response analyses. On the hardware side, microcontrollers like Arduino [5]-[18] have emerged as effective platforms for control, thanks to their compatibility with a range of programming languages and their ability to generate Pulse Width Modulation (PWM) signals for motor control. Control systems [19]-[22] typically employ techniques such as PID control [23]-[26] and Fuzzy Logic control [27]-[33] to optimize motor efficiency. Recent developments even include innovative algorithms like the Enhanced Artificial Transgender Longicorn Algorithm and Recurrent Neural Networks [34] for stabilizing Brushless DC (BLDC) motors. However, despite these advancements, there is a need for more sophisticated control algorithms that can better mimic human-like movements. Thus, our research aims to design, develop, and test a robotic arm modeled after human arm movements. The robotic arm is controlled using a combination of MATLAB/Simulink software and Arduino hardware, implementing Trajectory Control techniques to achieve the desired motion paths. Smooth Functions are also incorporated to minimize system overshoot and enhance accuracy.

In this research, the focus is on designing and developing a robotic arm that simulates the movements of a human arm. The robotic arm is equipped with specific functionalities, including shoulder lifting, lateral arm sweeping, and elbow folding and extending. MATLAB/Simulink is employed for controlling these movements, and various tests are conducted to assess the efficiency of the robot's mobility. The study aims to contribute to the field by enhancing the understanding and optimization of robotic arms that closely emulate human arm dynamics.

2. RESEARCH METHODOLOGY

This study encompasses various components such as hardware structure design, system control, current regulation through motors, and the circuits used in designing the robotic arm. The details are elaborated under the following subheadings.

2.1. Hardware Structure Design

In designing the hardware structure, CATIA software is utilized to model the robotic arm. This research aims to simulate human arm movements based on human anatomy, featuring a 4-axis movement system, as shown in Fig. 3.1. This system comprises 1) vertical shoulder movement, 2) lateral shoulder sweeping, 3) elbow folding and extending, and 4) a gripping hand capable of twisting in and out, as illustrated in Fig. 1.

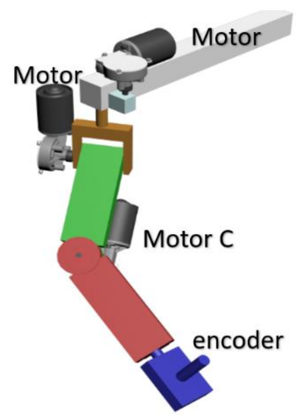


Figure 1. Depiction of various pivot points of the simulated robotic arm.

The complete setup, as discussed, operates in tandem as shown in Fig. 2. The movement is actuated by a total of three motors, referred to as Motors A, B, and C. Motor A controls the lateral shoulder sweeping, Motor B controls vertical shoulder movement, and Motor C simulates elbow folding and extending. The motors are governed by PWM signals from the microcontroller, directing electrical pulses to different motors. Once the motors rotate along the various axes, the encoder measures the angles and sends electrical feedback to the microcontroller. This data is then relayed to a computer for further analysis.

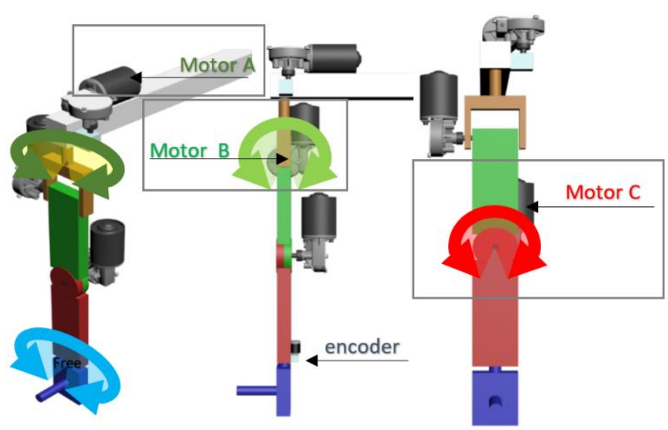


Figure 2. Display of different operational pivot points of the simulated robotic arm.

2.2. Electronics Design Section for Robotic Arm Control System

In the design of the robotic arm control system, the research comprises three major components for operational control.

The Primary Processing Unit involves a computer system utilizing MATLAB/Simulink software for reading various parameters of the robotic arm operation. This data is processed to direct the robot towards a specified target.

The Secondary Processing Unit serves as a signal encoder for data transmission between the computer and the microcontroller board. This unit is responsible for sending processed parameters from the computer to drive the motors, and for receiving sensor data to specify the positions that have been moved. This data is then sent back to the computer for further processing.

The Motor Control Unit consists of a motor driver circuit, a DC motor, and an adjustable resistor. This unit receives the command to operate the motor, directing it to various desired positions to achieve the specified objectives, as illustrated in Fig. 3.

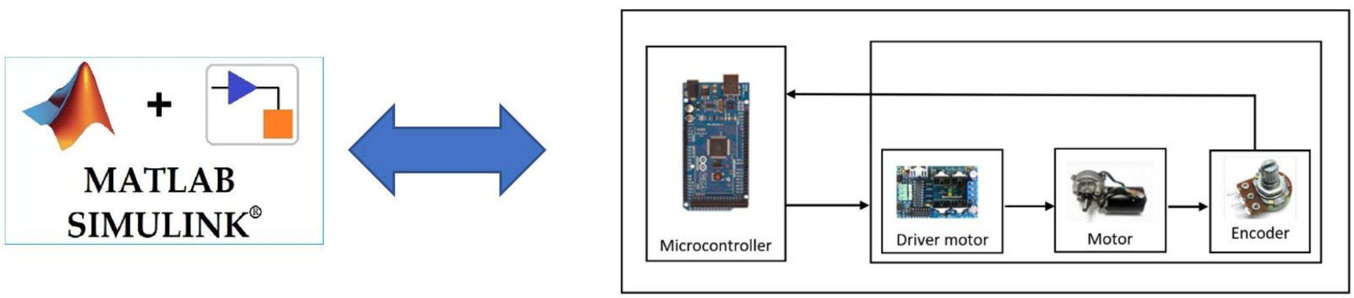


Figure 3. Demonstrates the Operation of the Simulated Robotic Arm.

2.3. Robotic Arm Gripping System

For object gripping control in this research, a Metal Robotic Arm Gripper equipped with optional servo motors MG995 or MG996R is selected. The system incorporates two servo motors that manage the gripping and twisting actions of the robotic arm. These motors operate under the control of an Arduino UNO microcontroller, which receives operational commands to drive the motors according to the desired program, as depicted in Fig. 4.

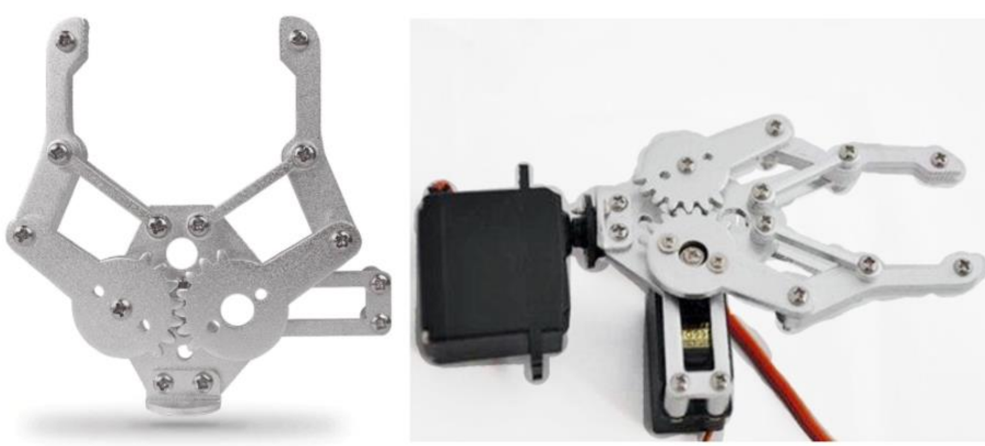


Figure 4. Illustrates the Functioning of the Simulated Robotic Arm.

2.4. Prototype Robotic Arm Implementation Results

The study and design phase of the robotic arm is depicted in Fig. 1. The arm is constructed using square steel tubing with dimensions of 25.4 x 25.4 x 2 mm. The assembly follows the design specifications and the prototype is showcased in Fig. 5.

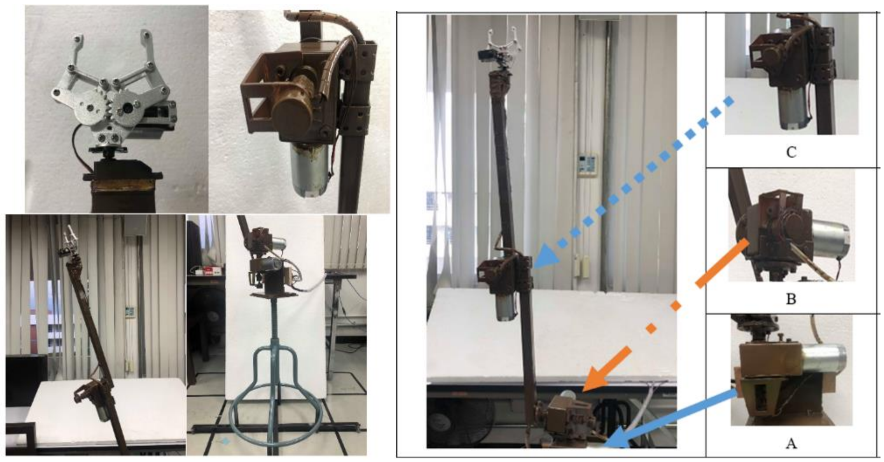


Figure 5. Illustrates the Designed and Assembled Robotic Structure.

The operation of the robotic arm involves three distinct motors. Motor A is positioned at the base of the arm and serves to rotate it through wide angles, simulating human shoulder movement. Motor B controls the arm's vertical lift, analogous to the upper arm movement in humans. Lastly, Motor C allows the robotic arm to perform lateral sweeping actions, resembling the human elbow joint. These functionalities are detailed in Fig. 5.

2.5. Designing a PID Control System for the Robotic Arm

The PID control system, short for Proportional-Integral-Derivative control, operates on closed-loop control systems, also known as feedback control systems. Modern automation has seen ongoing advancements in techniques and methods to improve the efficiency of continuous control systems. The PID controller, as shown in Fig. 6, remains the most widely accepted approach in industrial applications due to its straightforward structure, uncomplicated design, and adaptability across various control tasks.

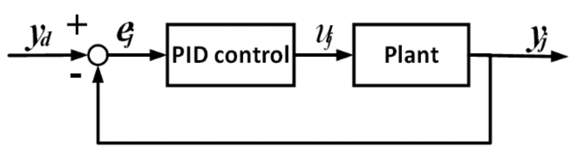


Figure 6. Block Diagram of the PID Control System.

The PID control mechanism comprises three sub controllers, 1) The Proportional term, or P-controller 2) The Integral term, or I-controller 3) The Derivative term, or D-controller. In regard to PID control theory, the framework exists in the form of a Continuous-Time PID controller. To implement this in a Microcontroller Unit (MCU), one needs to adapt it to a Discrete-Time PID controller, which can be derived from the continuous-time PID controller theory as expressed in Equation (1).

$$u(k) = K_p e(k) + K_i \sum_{0}^{k} e(k) + K_d \frac{e(k) - e(k-1)}{\Delta t} \tag{1}$$

2.6. Program Structure Design using MATLAB/Simulink

For designing the control system, the researchers have implemented PID control, dividing the control into three segments based on the motors in each axis. Trajectory Control is employed to dictate the motion paths over time. Within the program, there are setpoints represented by boxes named in_m1, in_m2, and in_m3, which define the desired motion paths for each motor. These setpoints are then fed into a Sub System box, containing the control system's set of commands. The results are displayed on a display screen and a scope screen, indicating the working status between the set values and the system's output, as shown in Fig. 7.

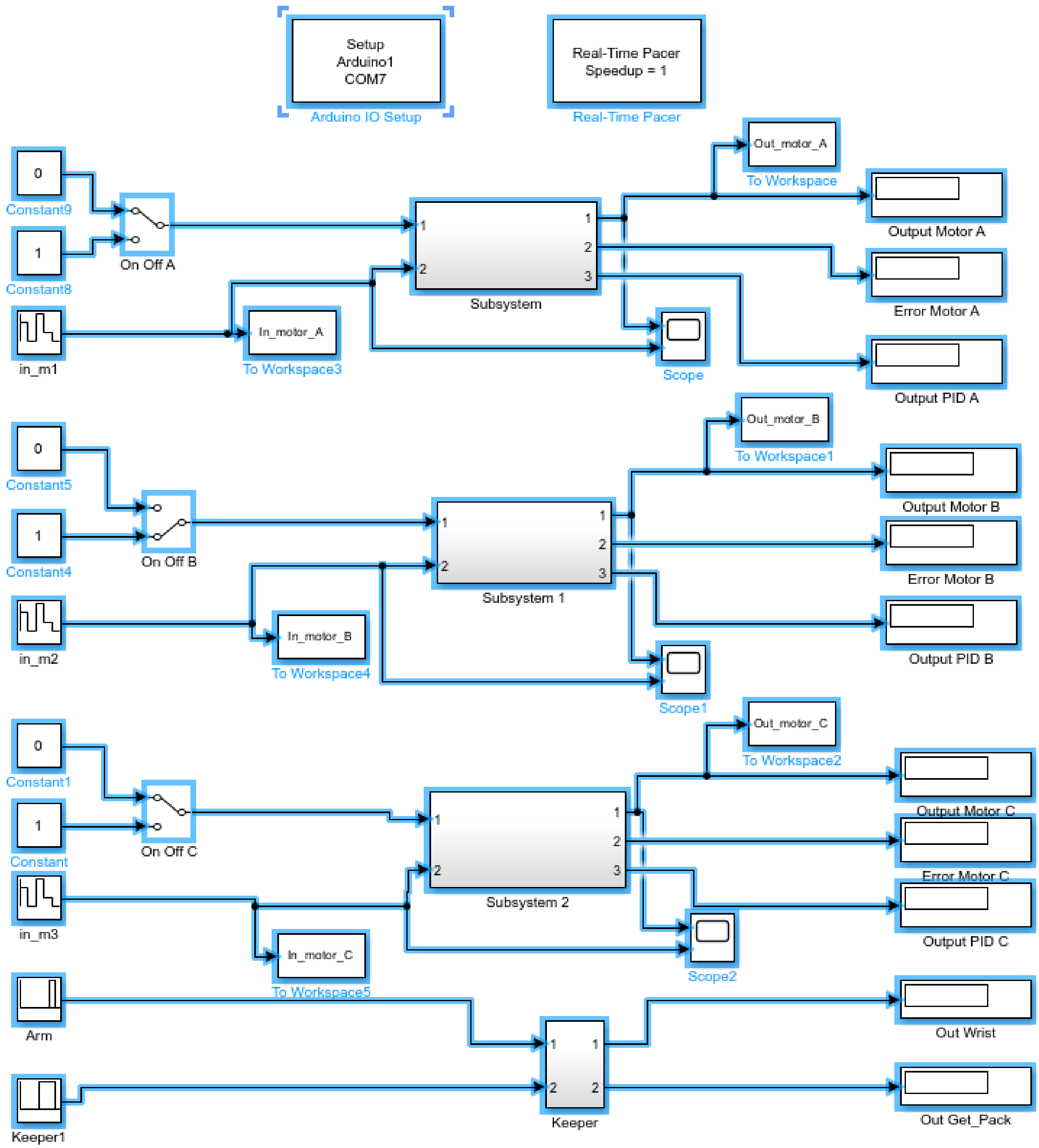


Figure 7. Illustration of the Program used for Trajectory Control.

Each Sub System in Fig. 7 has been encapsulated in a block diagram for ease of use. Examples are demonstrated, where the input data for control includes in_m1, in_m2, in_m3, Arm, and Keeper1. This displays the movements of the robotic arm as defined, illustrated in Fig. 8, which shows the control section and the conditions for setting the motors' movements, including verifying the positions of each motor.

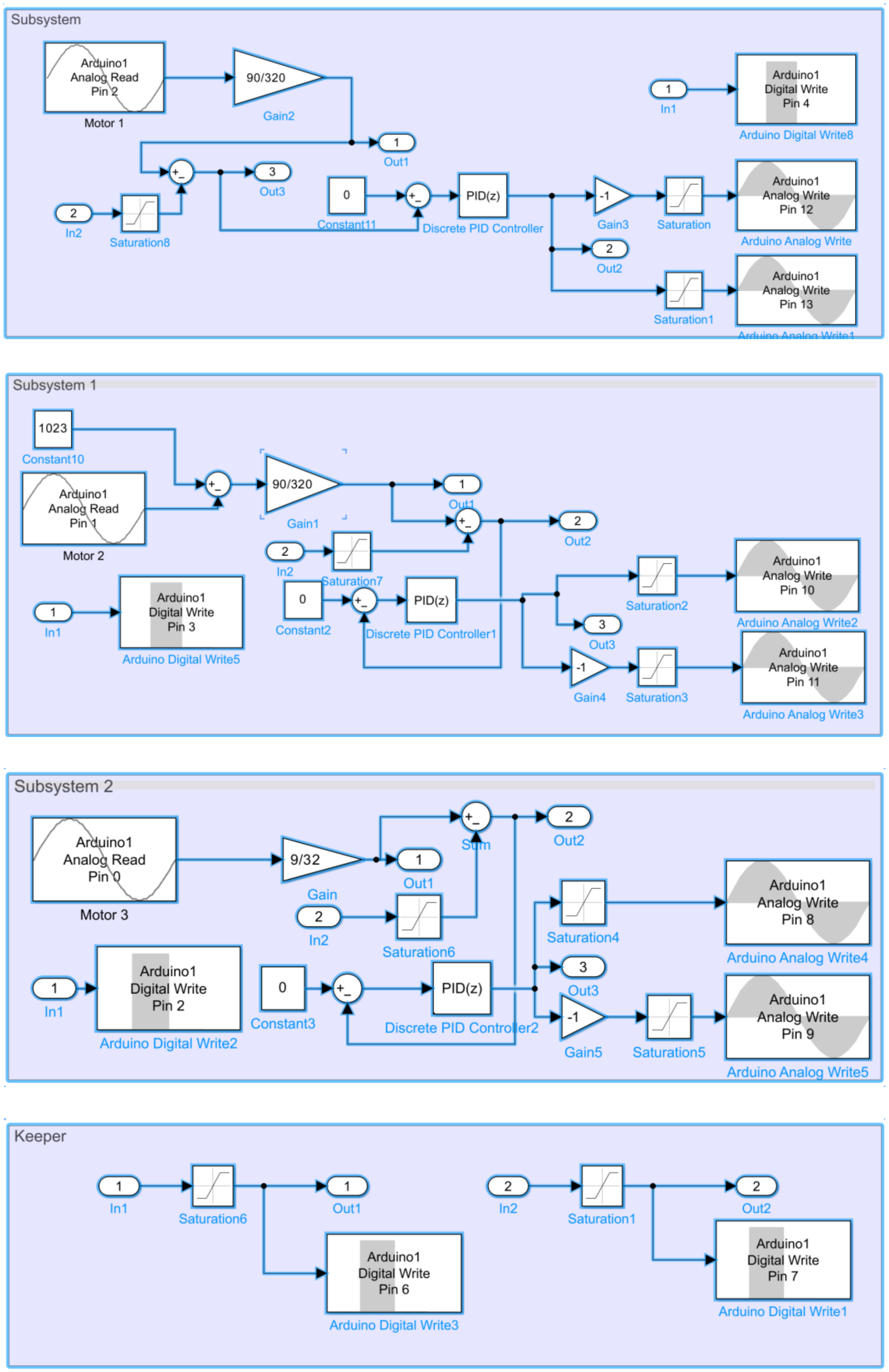


Figure 8. Example of Sub Systems in the Program used for Trajectory Control.

2.7. Defining the Motion Path for Robot Control

To control the robot's movements, the control program begins by setting the robot's coordinates to an initial system state. Specifically, the robot must return to its origin where the coordinates for axes A, B, and C are set to 0 degrees. The initial state values are set to continuously control the angles towards predetermined points and enable repetitive operation. This is possible through the use of the preliminary Simulink program, as shown in Fig. 7. In specifying the robot's movements, two types can be defined: Step input and smooth function. In this study, step input will be used to establish the motion path, as presented in Table 1. The system equations for each axis are applied, and a sampling time of 0.055 ms has been chosen for control system data collection.

Table 1. Table demonstrating the test results of the robot arm's step input

Process	Step time (k)		Setpoint A (Deg)	Setpoint B (Deg)	Setpoint C (Deg)
	Start (X)	End (Y)			
0	0	94	0	0	0
1	95	177	40	0	0
2	178	279	40	80	0
3	280	357	40	80	60
4	358	435	60	80	60
5	436	519	60	60	60
6	520	597	20	60	60
7	598	675	20	80	60
8	676	765	0	80	60
9	766	853	0	0	60
10	854	909	0	0	0

For the smooth function-based motion path, data is collected at a sampling time of 0.055 ms. The values determined for the step input from Table 2 are used to define the path. The path is approximated using the Smooth function equation, $S(x) = \frac{1}{1+e^{-x}}$, as per Equation 2. This enables the construction of a motion path based on the smooth function for system control.

$$S(x) = X + \frac{Y + X}{1 + e^{(-2(t-5))}} \tag{2}$$

3. RESULTS

In our experiments employing Trajectory Control, the motion paths were predefined as per Table 1 to test the robotic arm's ability to reach its target destination. Using the software depicted in Fig. 7 and the smooth function-based movement system, responses were recorded and are displayed in Fig. 9 - Fig11. These figures demonstrate the movements of motors A, B, and C according to the predefined angles listed in Table 2.

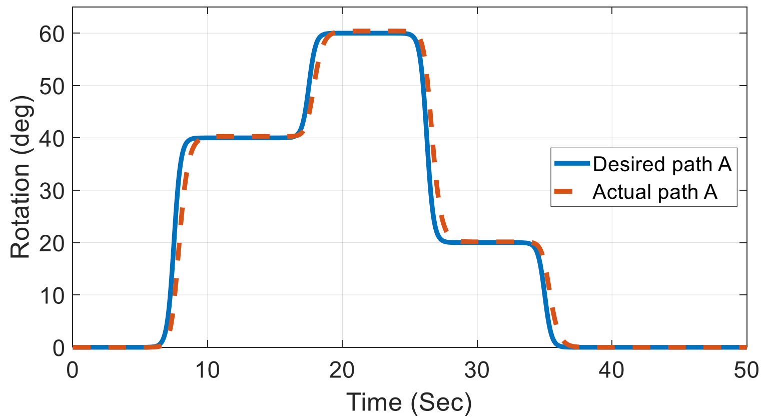


Figure 9. Displaying the Movement Results of the System in Motor Axis A.

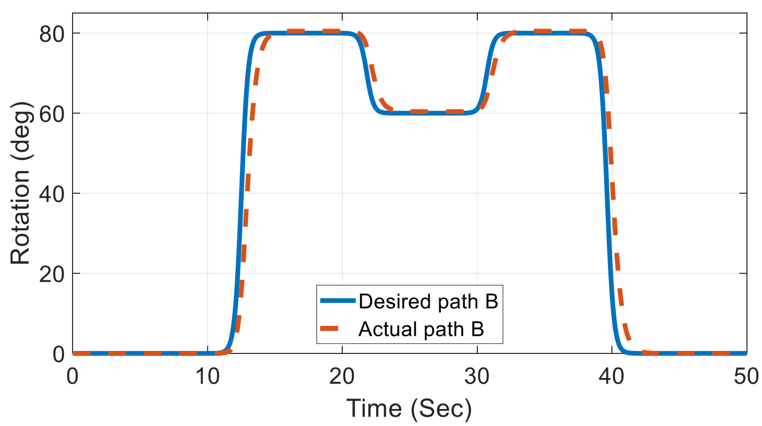


Figure 10. Displaying the Movement Results of the System in Motor Axis B.

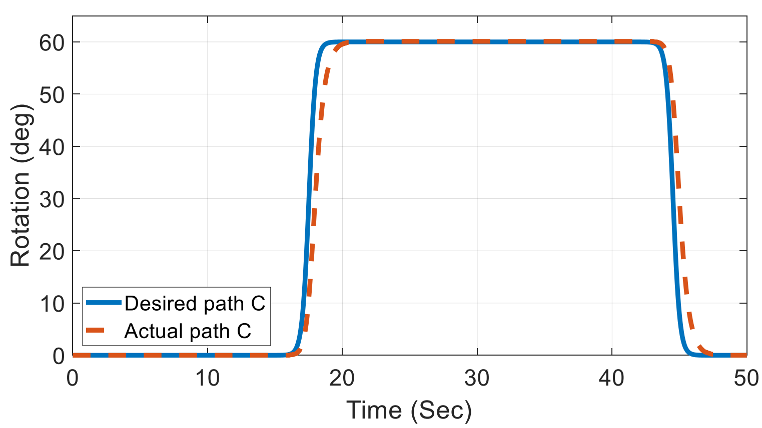
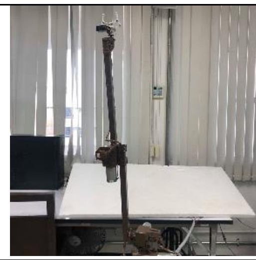
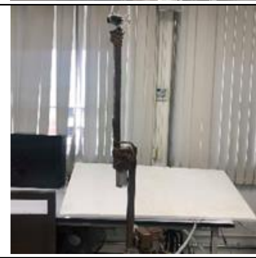


Figure 11. Displaying the Movement Results of the System in Motor Axis C.

From the data collected, which is presented in Table 2, the robotic arm effectively followed the predefined motion paths for each axis, as shown in Fig. 9- Fig. 11. Motor A, Motor B, and Motor C performed with RMS errors of 2.8587, 5.7340, and 4.4406, respectively, indicating the system's efficient control over movement.

Table 2. Table demonstrating the test results of the robot arm's step input

Step	Setpoint A	Setpoint B	Setpoint C	Illustrations of the Robot's Movements	Showcasing the Operations of the Robot Arm's Pivot Axis
0	0	0	0		Each axis is set to zero in the system as specified.
1	40	0	0		Motor A is set to 40 Deg to control the pivot at the shoulder position.
2	40	80	0		Motor B is set to 80 Deg to control the shoulder's downward movement.
3	40	80	60		Motor C is set to 60 Deg to control the elbow's folding inward.
4	60	80	60		Motor A is set to 60 Deg to control the shoulder pivot for inward rotation.
5	60	60	60		Motor B is set to 60 Deg to control the shoulder's upward movement.

6	20	60	60		Motor A is set to 20 Deg to control the shoulder pivot for outward rotation.
7	20	80	60		Motor B is set to 80 Deg to control the shoulder's downward movement again.
8	0	80	60		Motor A is set to 0 Deg to control the shoulder pivot for outward rotation.
9	0	0	60		Motor B is set to 0 Deg to control the shoulder's upward movement.
10	0	0	0		Motor C is set to 0 Deg to control the elbow's unfolding outward.

During the experiment, signal values were collected from each axis. The researchers used System Identification commands to approximate the transfer functions for each axis. The approximations were made in the Time Domain, using Discrete Time equations and a sampling time of 0.055 seconds. The transfer function equations for each axis are represented by Equations (3), (4), and (5).

$$\hat{G}_A(z) = \frac{0.072767z}{z^2 - 1.350753z + 0.423020} \tag{3}$$

$$\hat{G}_B(z) = \frac{0.065985z}{z^2 - 1.350992z + 0.416566} \tag{4}$$

$$\hat{G}_C(z) = \frac{0.059974z}{z^2 - 1.360473z + 0.420346} \tag{5}$$

CONCLUSIONS

This research focuses on designing and implementing a MATLAB/Simulink-based control system for a robotic assistant's arm, capable of various movements like lateral arm sweeping, arm folding and unfolding, vertical arm lifting, and object grasping. The design incorporates four pivot points to emulate human-like arm mechanics. Notable outcomes include the robot's ability to move according to user-defined coordinates and execute preset, time-dependent system commands. The robot can also mimic human anatomical arm movements and incorporates a PID control system for error compensation and system stability. Equations governing the robot arm's mechanics were also derived, providing a foundation for future development and computational simulations.

ACKNOWLEDGMENTS

The researchers would like to express their heartfelt gratitude to the Research Institute, Academic Services Center, and the College of Biomedical Engineering at Rangsit University for their generous financial support of this research. Special appreciation is also extended to Research Assistant Manutsawee Kiew-ong-art for her invaluable assistance in data collection. Additional thanks are due to our project advisors Nuntachai Thongpance, Anantasak Wongkamhang, and Anuchit Nirapai for their invaluable guidance and expertise.

REFERENCES

[1] G. Ren, "MATLAB/Simulink-Based Simulation and Experimental Validation of a Novel Energy Storage System to a New Type of Linear Engine for Alternative Energy Vehicle Applications," in IEEE Transactions on Power Electronics, vol. 33, no. 10, pp. 8683-8694, Oct. 2018, doi: 10.1109/TPEL.2017.2784563.

[2] A. H. El Din, Mohamed Ashraf Abdullah and M. Ibrahim, "A MATLAB/SIMULINK model to study the performance of the VFT for the interconnection of weak and strong AC grids," 2011 IEEE International Electric Machines & Drives Conference (IEMDC), 2011, pp. 1635-1640, doi: 10.1109/IEMDC.2011.5994607.

[3] A. Gastli, S. Kiranyaz, R. Hamila and O. Ellabban, "Matlab/Simulink Modeling and Simulation of Electric Appliances Based on their Actual Current Waveforms," 2019 2nd International Conference on Smart Grid and Renewable Energy (SGRE), 2019, pp. 1-7, doi: 10.1109/SGRE46976.2019.9020901.

[4] D. Saputra, A. Ma'arif, H. Maghfiroh, P. Chotikunnan, and S. Rahmadhia, "Design and Application of PLC-based Speed Control for DC Motor Using PID with Identification System and MATLAB Tuner," Int. J. Robotics and Control Systems, vol. 3, no. 2, pp. 233-244, Apr. 2023, DOI: 10.31763/ijrcs.v3i2.775.

[5] J. Sheng, "Teaching Devices and Controls for Computer Engineering and Systems Students using Arduino and MATLAB/Simulink," 2018 IEEE 14th International Conference on Control and Automation (ICCA), 2018, pp. 318-323, doi: 10.1109/ICCA.2018.8444192..

[6] M. Gavran, M. Fruk and G. Vujisić, "PI controller for DC motor speed realized with Arduino and Simulink," 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017, pp. 1557-1561, doi: 10.23919/MIPRO.2017.7973669.

[7] S. P. Biswas, M. K. Hosain and M. W. Rahman, "Real-time arduino based simulator enabled hardware-in-the-loop electric DC machine drive system," 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), 2017, pp. 823-826, doi: 10.1109/R10-HTC.2017.8289082.

[8] A. Ma'arif and N. R. Setiawan, "Control of DC motor using integral state feedback and comparison with PID: simulation and arduino implementation," Journal of Robotics and Control (JRC), vol. 2, no. 5, pp. 456-461, 2021, DOI: 10.18196/jrc.25122.

[9] J. Susilo, A. Febriani, U. Rahmalisa, and Y. Irawan, "Car Parking Distance Controller Using Ultrasonic Sensors Based On Arduino Uno," Journal of Robotics and Control (JRC), vol. 2, no. 5, pp. 353-356, 2021, doi: https://doi.org/10.18196/jrc.25106.

[10] N. Thongpance and P. Chotikunnan, "Design and Construction of Electric Wheelchair with Mecanum Wheel," Journal of Robotics and Control (JRC), vol. 4, no. 1, pp. 71-82, 2023, doi: https://doi.org/10.18196/jrc.v4i1.17095.

[11] A. Ma'arif and A. Çakan, "Simulation and Arduino Hardware Implementation of DC Motor Control Using Sliding Mode Controller," Journal of Robotics and Control (JRC), vol. 2, no. 6, pp. 582-587, 2021, doi: https://doi.org/10.18196/jrc.26140.

[12] R. Perkasa, R. Wahyuni, R. Melyanti, H. -, and Y. Irawan, "Light Control Using Human Body Temperature Based on Arduino Uno and PIR (Passive Infrared Receiver) Sensor," Journal of Robotics and Control (JRC), vol. 2, no. 4, pp. 307-310, 2021, doi: https://doi.org/10.18196/jrc.2497.

[13] Y. Irawan, A. Febriani, R. Wahyuni, and Y. Devis, "Water Quality Measurement and Filtering Tools Using Arduino Uno, PH Sensor and TDS Meter Sensor," Journal of Robotics and Control (JRC), vol. 2, no. 5, pp. 357-362, 2021, doi: https://doi.org/10.18196/jrc.v2i5.10166.

[14] Y. Irawan, M. Muhardi, R. Ordila, and R. Diandra, "Automatic Floor Cleaning Robot Using Arduino and Ultrasonic Sensor," Journal of

Robotics and Control (JRC), vol. 2, no. 4, pp. 240-243, 2021, doi: https://doi.org/10.18196/jrc.2485.

[15] Y. Irawan, R. Wahyuni, and H. Fonda, "Folding Clothes Tool Using Arduino Uno Microcontroller And Gear Servo," Journal of Robotics and Control (JRC), vol. 2, no. 3, pp. 170-174, 2021, doi: https://doi.org/10.18196/jrc.2373.

[16] A. Al Tahtawi, M. Agni, and T. Hendrawati, "Small-scale Robot Arm Design with Pick and Place Mission Based on Inverse Kinematics," Journal of Robotics and Control (JRC), vol. 2, no. 6, pp. 469-475, 2021, doi: https://doi.org/10.18196/jrc.26124.

[17] A. Mohammed, A. Abdullahi, and A. Ibrahim, "Development of a Prototype Autonomous Electric Vehicle," Journal of Robotics and Control (JRC), vol. 2, no. 6, pp. 559-564, 2021, doi: https://doi.org/10.18196/jrc.26137.

[18] P. Anupongongarch, T. Kaewgun, J. A. O'Reilly, and S. Suraamornkul, "Design and construction of a non-invasive blood glucose and heart rate meter by photoplethysmography," Journal of Current Science and Technology, vol. 12, no. 1, pp. 89–101, 2023. [Online]. Available: https://ph04.tci-thaijo.org/index.php/JCST/article/view/327

[19] J. Liu, F. Wang and P. Zhang, "Real-Time DC Servo Motor Position Control by PID Controller Using Labview," in 2009 International Conference on Intelligent Human-Machine Systems and Cybernetics. IHMSC 2009, Hangzhou, Zhejiang, 2009 pp. 206-209, doi: 10.1109/IHMSC.2009.59

[20] K. Kirkpatrick and G. Chaudhry, "DC Machine Control with Time Delay and a PID Controller," in 3rd Annual IEEE Conference on Automation Science and Engineering, Scottsdale, AZ, 2007 .doi: 10.1109/COASE.2007.4341753

[21] M. Fallahi and S. Azadi, "Fuzzy PID Sliding Mode Controller Design for the Position Control of a DC Motor," in 2009 International Conference on Education Technology and Computer, ICETC 2009, Singapore, 2009 pp. 73-77, doi: 10.1109/ICETC.2009.12

[22] W. Na, "A New Fuzzy PID Controller for Time Delay Systems," in 2009 WRI Global Congress on Intelligent Systems (GCIS), Xiamen, 2009 pp. 191-194. doi: 10.1109/GCIS.2009.122

[23] M. Rabah, A. Rohan, and S. H. Kim, "Comparison of position control of a gyroscopic inverted pendulum using PID, fuzzy logic and fuzzy PID controllers," International Journal of Fuzzy Logic and Intelligent Systems, vol. 18, no. 2, pp. 103–110, 2018, doi: 10.5391/IJFIS.2018.18.2.103.

[24] D. Tran, N. Hoang, N. Loc, Q. Truong, and N. Nha, "A Fuzzy LQR PID Control for a Two-Legged Wheel Robot with Uncertainties and Variant Height," Journal of Robotics and Control (JRC), vol. 4, no. 5, pp. 612–620, 2023, doi: 10.18196/jrc.v4i5.19448.

[25] A. Latif, A. Zuhri Arfianto, H. Agus Widodo, R. Rahim, and E. T. Helmy, "Motor DC PID System Regulator for Mini Conveyor Drive Based-on Matlab," Journal of Robotics and Control (JRC), vol. 1, no. 6, pp. 185-190, 2020, DOI: 10.18196/jrc.1636.

[26] P. Chotikunnan and R. Chotikunnan, "Dual Design PID Controller for Robotic Manipulator Application," Journal of Robotics and Control (JRC), vol. 4, no. 1, pp. 23-34, 2023, doi: https://doi.org/10.18196/jrc.v4i1.16990.

[27] B. Panomruttanarug and P. Chotikunnan, "Self-balancing iBOT-like wheelchair based on type-1 and interval type-2 fuzzy control," in 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), May 2014, pp. 1-6, DOI: 10.1109/ECTICon.2014.6839710.

[28] P. Chotikunnan and B. Panomruttanarug, "The application of fuzzy logic control to balance a wheelchair," Journal of Control Engineering and Applied Informatics, vol. 18, no. 3, pp. 41-51, 2016.

[29] P. Chotikunnan, R. Chotikunnan, A. Nirapai, A. Wongkamhang, P. Imura, and M. Sangworasil, "Optimizing Membership Function Tuning for Fuzzy Control of Robotic Manipulators Using PID-Driven Data Techniques," Journal of Robotics and Control (JRC), vol. 4, no. 2, pp. 128-140, 2023, DOI: 10.18196/jrc.v4i2.18108.

[30] P. Chotikunnan, and Y. Pititheeraphab, " Adaptive P Control and Adaptive Fuzzy Logic Controller with Expert System Implementation for Robotic Manipulator Application," Journal of Robotics and Control (JRC), vol. 4, no. 2, pp. 217-226, 2023.

[31] M. K. Saleem, M. L. U. R. Shahid, A. Nouman, H. Zaki, and M. A. U. R. Tariq, ""Design and implementation of adaptive neuro-fuzzy inference system for the control of an uncertain ball and beam apparatus,"" Mehran University Research Journal Of Engineering & Technology, vol. 41, no. 2, pp. 178-184, 2022, DOI: 10.22581/muet1982.2202.17."

[32] P. Chotikunnan, B. Panomruttanarug, N. Thongpance, M. Sangworasil, and T. Matsuura, ""An application of Fuzzy Logic Reinforcement Iterative Learning Control to Balance a Wheelchair,"" International Journal of Applied Biomedical Engineering, vol. 10, no. 2, pp. 1–9, 2017."

[33] R. Chotikunnan, P. Chotikunnan, A. Ma'arif, N. Thongpance, Y. Pititheeraphab, and A. Srisiriwat, ""Ball and Beam Control: Evaluating Type-1 and Interval Type-2 Fuzzy Techniques with Root Locus Optimization,"" International Journal of Robotics and Control Systems, vol. 3, no. 2, pp. 286–303, 2023, doi: 10.31763/ijrcs.v3i2.997."

[34] P. Rajesh, F. H. Shajin, V. Ansal, and V. K. B., "Enhanced artificial transgender longicorn algorithm & recurrent neural network based enhanced DC-DC converter for torque ripple minimization of BLDC motor," Journal of Current Science and Technology, vol. 13, no. 2, pp. 182–204, 2023, doi: https://doi.org/10.59796/jcst.V13N2.2023.1735.

DOI: https://doi.org/10.15379/ijmst.v10i3.1974