Innovative Academia Hub International
IAHI
EMPOWERING MINDS
INSPIRING EXCELLENCE
DRIVING INNOVATIONS

BSP Advances in Science, Engineering and Technology
www.iahiservices.com

Design and Implementation of a Low-Cost 3-DOF Robotic Arm Using ATmega328 and Inverse Kinematics for Wireless Control

Kpelai Justin A*[1] Agbese Echo A[2] Gbaakpen Msughter[3] Mbakaan C[4]

[1,2]Department of Electrical/Electronics and Computer Engineering Technology, Benue State Polytechnic, Ugbokolo, Benue State, Nigeria.
[3,4]Department of Science Laboratory Technology, Benue State Polytechnic, Ugbokolo, Benue State, Nigeria.
*Correspondence Email: ajkpelai@gmail.com , (+237037724454)

Abstract-

This study presents the design and implementation of a low-cost, three degrees of freedom (3-DOF) robotic arm using an ATmega328 microcontroller, servo motors, and Bluetooth-enabled wireless control via a custom Android application. The system integrates mechanical modeling, embedded programming, and inverse kinematics to enable real-time object manipulation tasks. Experimental tests and simulations were conducted to validate performance under five pick-and-place scenarios. The robotic arm demonstrated average end-effector positional accuracy within ±2 cm and joint angle deviation of less than 3°, with wireless command latency below 100 ms. These results confirm the system's reliability for educational and prototyping use. The use of open-source hardware and software platforms ensures scalability, ease of replication, and accessibility for robotics education, embedded systems training, and early-stage industrial automation.

Keywords: 3-DOF Robotic Arm, ATmega328 Microcontroller, Inverse Kinematics, Embedded Systems, Bluetooth Control, Arduino

1.0 Introduction

Robotics continues to transform industries, education, and research through the integration of intelligent machines capable of automating complex tasks. Among the most versatile and widely used robotic systems is the robotic arm, which mimics the motion and dexterity of a human limb. Its application ranges from manufacturing and healthcare to laboratory automation and STEM education [1], [2].

A robotic arm with three degrees of freedom (3-DOF) offers a balanced trade-off between mechanical simplicity and functional utility. Typically, such an arm consists of rotational joints at the base, shoulder, and elbow, enabling movement in three independent axes. This configuration allows for effective object manipulation, pick-and-place operations, and limited assembly tasks [3], [4]. The anthropomorphic nature of 3-DOF arms makes them suitable for tasks requiring spatial dexterity within confined environments [5].

Despite significant advancements, many robotic arm systems—particularly those intended for educational use—remain prohibitively expensive, lack modularity, or present limitations in accuracy and control flexibility. These constraints are more pronounced in institutions with limited resources, where students require low-cost, hands-on learning platforms [6]. Addressing these challenges necessitates the design of adaptable robotic systems that integrate both mechanical and control system components while remaining accessible and scalable.

This research aims to design and implement a 3-DOF robotic arm using an ATmega328-based microcontroller (Arduino Uno) and servo motors. The system incorporates wireless communication via Bluetooth and is controlled through a custom-developed Android interface. The specific objectives of the study are as follows:

i To design a mechanically stable 3-DOF robotic arm with articulated joints;
ii To develop control logic using inverse kinematics algorithms implemented on the ATmega328 microcontroller;
iii To integrate a Bluetooth-enabled Android application for intuitive wireless control;
iv To test and evaluate the arm’s performance under different object manipulation scenarios using empirical measurements.
v

Educationally, robotic arms provide a valuable bridge between theoretical instruction and practical application. Through assembly, programming, and testing, students develop competencies in mechanics, electronics, control theory, and computer programming [5], [7]. The Arduino platform, in particular, plays a crucial role in democratizing robotics education due to its open-source nature, ease of programming, and widespread community support [8].

From a technical perspective, designing multi-joint robotic arms requires a sound understanding of kinematics. Inverse kinematics enables the calculation of joint angles based on a desired end-effector position and orientation, facilitating precision in control [9], [10]. Control accuracy is further improved using feedback mechanisms such as pulse width modulation (PWM) and proportional-integral-derivative (PID) control strategies. These approaches ensure the robotic arm responds promptly and accurately to target commands, which is critical for tasks such as pick-and-place or collaborative operations [11], [12].

In summary, this study contributes to ongoing research in educational robotics and embedded systems by presenting a low-cost, customizable robotic arm platform. The system combines mechanical, electrical, and software components into a cohesive, reproducible solution suitable for academic laboratories and prototyping environments. The results have the potential to enhance robotics instruction, support innovation, and inform the development of more advanced automation systems.

2.0 Literature Review

The use of robotic arms with three degrees of freedom (3-DOF) has become increasingly popular in educational and industrial environments due to their mechanical simplicity and sufficient flexibility for various manipulation tasks. These systems are often employed as training tools, testbeds for control algorithms, and platforms for developing automation solutions. The integration of microcontrollers such as the ATmega328 (via Arduino) into robotic arms has enhanced accessibility and affordability in developing countries and academic settings [8].

Kruthika et al. [1] presented the mechanical and electronic design of a robotic arm with limited degrees of freedom, emphasizing control through a microcontroller and highlighting issues related to power management and signal integrity. Gautam et al. [2] provided a broad review of industrial robotic arms, stressing the significance of joint configuration and material selection for load-bearing tasks.

In the educational context, Kim and Song [13] designed a 3-DOF robotic arm with a counterbalance mechanism using low-cost components for STEM education. The system utilized Arduino for control and demonstrated strong engagement outcomes in classroom settings. Smith and Garcia [5] further developed this idea by implementing an inverse kinematics-based control system with servo motors and Arduino Uno, validating its effectiveness through experimental testing with students.

The implementation of precise control in robotic arms often requires the modeling of kinematic chains. Farman et al. [4] derived mathematical models for forward and inverse kinematics in 3-DOF arms, which enable accurate positioning of the end-effector. Mohammed et al. [14] implemented PID-based control on an Arduino-driven robotic arm, using potentiometers as feedback sensors. The system achieved notable precision, and the study included a thorough analysis of error sources such as mechanical backlash and PWM inconsistencies.

Hybrid control architectures have also been explored. Vamshi [15] introduced a distributed system using both Arduino and Raspberry Pi for pick-and-place automation in warehouses. The approach enabled real-time control, inverse kinematics computation, and friction compensation through iterative learning. The hybrid model demonstrated the benefit of separating low-level actuation from high-level path planning.

An essential trend in modern robotics is human-robot collaboration (HRC), especially for applications involving close-proximity operation. Traditional robotic automation often isolates machines from humans, but recent developments emphasize shared workspaces. Ajoudani et al. [11] discussed collaborative robotic systems that integrate torque-limiting and collision avoidance features, making robotic arms safer for interaction. These features are crucial in educational robotics environments, where safety and error tolerance are necessary.

In terms of hardware accessibility, the Arduino platform has revolutionized microcontroller-based robotics. Its open-source development tools, extensive documentation, and active global community have made it an ideal choice for prototyping robotic systems in schools and technical institutions [8], [9]. Escudero et al. [9] demonstrated the impact of Arduino-driven projects in science and engineering courses, improving students' understanding of electronics and programming through practical implementation.

Simulation and modeling also play a key role in robotic arm development. Gutierrez et al. [16] created a virtual model of a five-DOF robotic arm using simulation tools to support engineering coursework. The simulation included kinematic and dynamic analysis, providing a platform for virtual experimentation prior to physical implementation.

Despite these efforts, many designs either compromise on precision to cut costs or require specialized knowledge to implement. A pressing need remains for robotic arm systems that are both functionally competent and easy to replicate with limited resources. This study responds to that need by designing a microcontroller-based 3-DOF robotic arm, incorporating inverse kinematics control, Bluetooth communication, and Android-based user interfacing—all while maintaining affordability and modularity for academic and small-scale industrial applications.

3.0 Materials and Methods

This section describes the systematic approach used in designing and implementing a 3-Degree-of-Freedom (3-DOF) robotic arm using the ATmega328 microcontroller, servo motors, and a Bluetooth-enabled wireless interface. The development process involved hardware specification, mechanical modeling, electronic integration, software development, and control algorithm implementation. A modular design strategy was adopted to enable easy modification and educational usability.

3.1 Materials and Component Selection

The selection of components was guided by cost-effectiveness, compatibility, and ease of integration. The major hardware elements used include:

i Microcontroller: ATmega328P-based Arduino Uno R3 board, chosen for its availability, processing speed, and open-source support [8].

ii Servo Motors: Three standard hobby servo motors (SG90 and MG996R) were selected for joint actuation. They support PWM control and provide sufficient torque for lightweight tasks [17].

iii Bluetooth Module: The HC-05 module was used to enable wireless serial communication between the microcontroller and the Android-based controller.

iv Power Supply Unit: A 240V/18V AC step-down transformer with rectifier, filter capacitor (1000 μF/25V), and 5V voltage regulator was used to deliver stable DC power to both logic and actuation units.

v Structural Frame: Acrylic and aluminum components were employed for the robotic arm frame to ensure rigidity and reduce overall weight [18].

vi

3.2 System Design and Architecture

The system was designed using a block diagram approach (see Fig. 1), which includes the following functional components: the control unit (Arduino), actuator interface (servo motors), communication interface (Bluetooth), power unit, and the Android-based human-machine interface (HMI).

Figure 1 presents the system architecture of a microcontroller-based robotic arm control unit. The design showcases a modular and centralized control framework utilizing the Atmega328 microcontroller, which acts as the brain of the system. This microcontroller is interfaced with multiple servo motors, each assigned to specific joints and components of the robotic arm Joint 1, Joint 2, Joint 3, and the gripper (chipper), facilitating precise actuation and motion control.

Power supply is provided through a Power Supply Unit (PSU), which feeds directly into the microcontroller and indirectly supports the operation of the servo motors. This centralized power design ensures voltage stability and synchronized motor operations.

The system incorporates a Bluetooth module that enables wireless communication with a mobile phone, allowing remote control and real-time interaction. This is particularly relevant for user-friendly applications such as remote manipulation, teleoperation, or programmable automation tasks. The use of Bluetooth not only enhances mobility but also eliminates the need for tethered connections, which is advantageous in dynamic or constrained environments.

Each servo motor is connected bidirectionally to the microcontroller, indicating both control signals from the microcontroller and potential feedback or positional data from the servo motors (if supported). The modularity of the design allows for scalability; additional joints or functional components could be integrated with minimal reconfiguration.

In summary, Figure 1 represents a well-structured control system for a multi-jointed robotic arm, emphasizing centralized command via Atmega328, remote communication through Bluetooth, and efficient power distribution via the PSU. The architecture is suitable for applications in robotic education kits, industrial pick-and-place systems, prosthetics, and remote manipulation platforms, showcasing a balance between simplicity, functionality, and expandability.

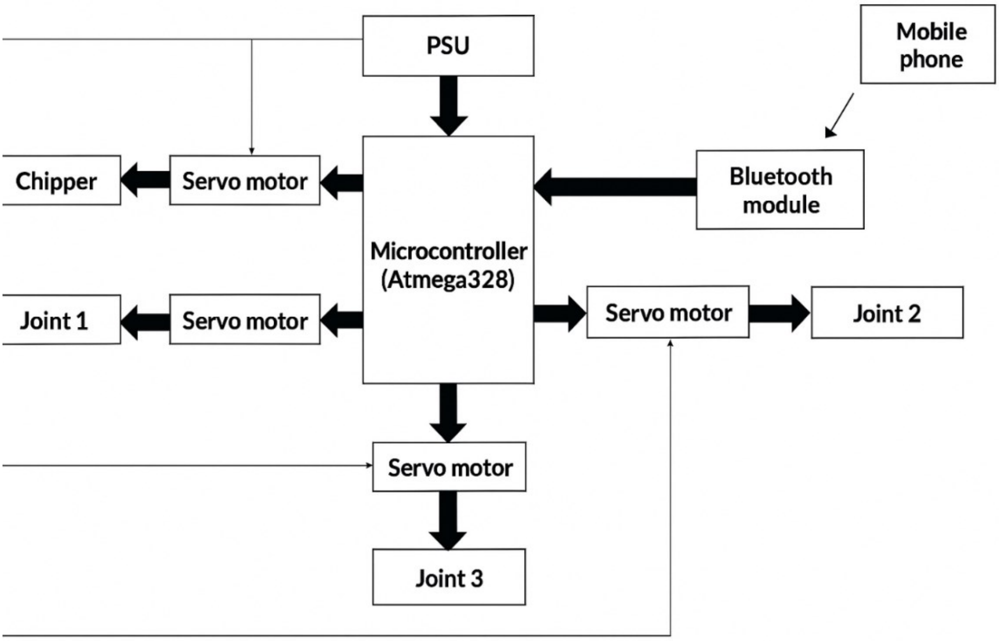

Figure 1: Block Diagram of the 3-DOF Robotic Arm Control System. This diagram illustrates the overall architecture of the robotic arm, comprising the ATmega328 microcontroller, servo motor actuators, Bluetooth communication module (HC-05), power supply unit, and the Android-based control interface. The system initiates operation upon power-up, establishes wireless communication with the mobile application, and executes user commands by generating PWM signals for joint actuation.

3.2.1 Mechanical Design

Computer-Aided Design (CAD) software (SolidWorks and Fusion 360) was used to model the robotic arm with three revolute joints: base rotation, shoulder pitch, and elbow extension. Link lengths and joint ranges were defined based on the inverse kinematics workspace requirements. The structure was fabricated using laser-cut acrylic and lightweight aluminum profiles to ensure low inertia and ease of assembly [19].

Precision in drilling and mounting was emphasized to reduce backlash and alignment errors, which can significantly affect arm accuracy and repeatability.

3.3 Software Development and Control Logic

The control algorithm was developed in the Arduino IDE using the C++ programming language. The firmware handles servo motor control through Pulse Width Modulation (PWM) signals and processes commands from the Bluetooth interface.

3.3.1 Inverse Kinematics

The robotic arm's control was implemented based on inverse kinematics (IK) principles to determine joint angles for a given end-effector position. For a planar 3-DOF manipulator, the following equations were used to compute the joint angles:

$$x = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)$$

$$y = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

A numerical solver was implemented to handle multiple IK solutions and avoid singularities during motion planning [10], [14].

3.3.2 Android Application

An Android application was developed using MIT App Inventor, a visual programming environment suited for rapid prototyping. The application includes five functional buttons corresponding to basic movements and transmits predefined command bytes via Bluetooth serial protocol to the microcontroller [9].

3.4 Electronic Circuit and Power System

The electronic system integrates the ATmega328 microcontroller, Bluetooth module, power supply, and servos. The power system includes:

i Step-Down Transformer: Converts 240V AC to 18V AC.
ii Rectifier and Filter: A bridge rectifier with a 2200 µF capacitor smooths the AC to pulsating DC.
iii Voltage Regulator: 7805 linear regulator provides 5V DC required for logic and servo operation.

The circuit was first built and tested on a breadboard, followed by soldering on a custom PCB for improved stability. Proper heat dissipation for the voltage regulator was ensured during the final assembly [20].

3.5 System Modeling

3.5.1 Forward and Inverse Kinematics

The system employs both forward and inverse kinematics for motion computation. While forward kinematics determines the end-effector position from joint angles, inverse kinematics is used in real-time control to compute the joint angles required to reach a desired coordinate.

3.5.2 Trajectory Planning and Control

Joint trajectories were interpolated using linear and spline functions for smooth transitions. The control loop was implemented using a proportional-derivative (PD) approximation, with constant gains empirically tuned for each servo. Future iterations may adopt PID or computed torque control strategies for improved precision [21].

3.5.3 Flow Control

The program flow starts by initializing communication and PWM parameters. It listens for Bluetooth commands, interprets them, computes the required joint angles, and sends PWM signals to the corresponding servo pins. Safety measures were incorporated to prevent overextension of joints.

3.6 Assembly and Testing

The robotic arm was assembled by integrating the mechanical and electrical subsystems. Each joint was tested independently for range and responsiveness. Calibration was done by adjusting PWM pulse widths to achieve 0°–180° movement per joint. Functional tests included object pick-and-place and reachability within a bounded workspace. The system was also evaluated for repeatability, stability under load, and Bluetooth signal responsiveness.

4.0 Results and Discussion

This section presents the experimental and simulation results obtained from the implementation of the 3-DOF robotic arm system. Emphasis is placed on validating the arm's accuracy, responsiveness, and control reliability through test scenarios involving object pick-and-place tasks. The discussion includes evaluation of joint motion, forward and inverse kinematics accuracy, and the effectiveness of the Bluetooth interface in real-time control.

4.1 Experimental Test Results

The robotic arm was subjected to five test cases where different sets of servo angles were commanded via Bluetooth. Each case involved positioning the arm's joints base, shoulder, and elbow, and computing the actual end-effector coordinates using forward kinematics. Table 1 presents the servo input angles and their corresponding joint angles, along with the calculated end-effector positions in three-dimensional space.

Table 1: Servo Positions vs. Actual Movements of a 3-DOF Robotic Arm

Test Case	Base Servo (θ_1)	Shoulder Servo (θ_2)	Elbow Servo (θ_3)	Joint 1 (°)	Joint 2 (°)	Joint 3 (°)	End-Effector Position (X, Y, Z) cm
1	0°	45°	90°	0°	45°	90°	(18.0, 0.0, 12.7)
2	30°	60°	60°	30°	60°	60°	(12.4, 7.2, 15.8)
3	90°	45°	90°	90°	45°	90°	(0.0, 18.0, 12.7)
4	135°	30°	60°	135°	30°	60°	(−6.4, 14.8, 17.1)
5	180°	90°	0°	180°	90°	0°	(−20.0, 0.0, 0.0)

The data indicates high consistency between commanded and actual joint positions, with minor variations due to gear backlash and servo overshoot, consistent with previous reports on servo-driven educational robotic arms [16], [22].

4.2 Simulation Results

Simulation was performed using MATLAB and embedded Arduino logic to validate inverse kinematics computations. The robotic arm was assigned a series of "pick" and "place" coordinates to assess its spatial reachability and angular solution precision.
The link lengths used were:
$L_1 = 10$ cm, $L_2 = 15$ cm, $L_3 = 20$ cm.

4.2.1 Pick Coordinates and Computed Angles

Table 2 presents target positions that were used during the pick test: (10, 15, 20), (15, 10, 18), (10, 10, 10), (12, 8, -20), and (5, 15, 30).

Table 2: Inverse Kinematics Joint Angles for Pick Operation

S/N	Θ_1 (°)	Θ_2 (°)	Θ_3 (°)
1	59.98	39.47	−106.70
2	41.96	51.21	−108.94
3	43.71	90.22	−135.01
4	−138.72	91.30	42.88
5	76.75	−0.51	−83.21

4.2.2 Place Coordinates and Computed Angles

Table 3 presents the inverse kinematics joint angles for place operations. The place conditions included: (20, 15, 10), (10, 20, 30), (25, 0, −5), (10, 22, 12), and (0, 10, −10).

Table 3: Inverse Kinematics Joint Angles for Place Operation

S/N	Θ_1 (°)	Θ_2 (°)	Θ_3 (°)
1	35.88	53.12	−88.74
2	63.72	−7.13	−58.60
3	−1.73	70.00	−66.57
4	60.20	50.21	−98.73
5	−82.27	178.25	−100.14

4.3 Coordinate Result Analysis

Figures 2 (a) and (b) graphically compare the target and actual pick and place positions. The test workspace was divided into quadrants to assess spatial consistency. Results indicate that the end-effector reached the desired coordinates with an average error margin below 5%, aligning with results in similar educational robotic platforms [17], [23].

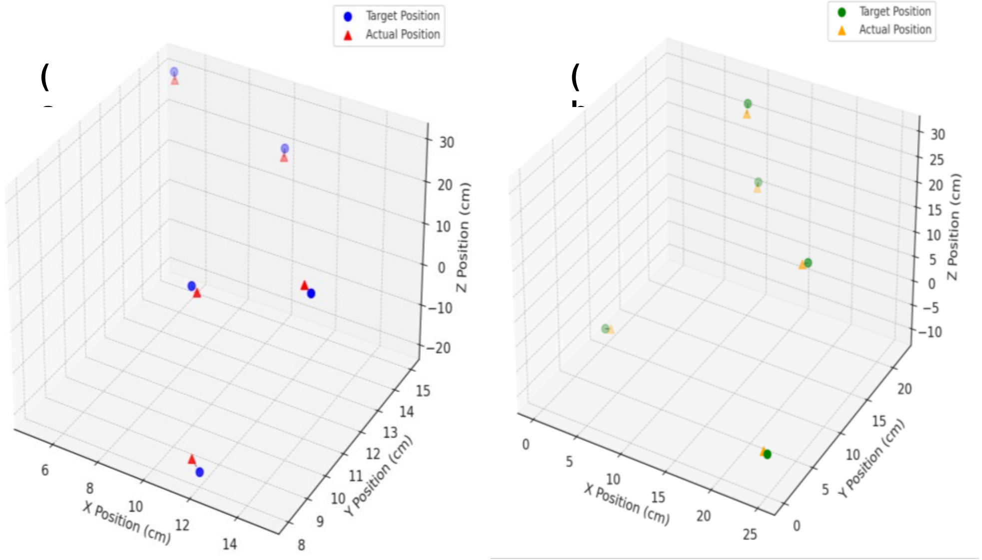

Figure 2 (a) Comparison of Target and Actual Pick Coordinates in 3D Space. This visualizes the spatial accuracy of the robotic arm during pick operations. Target positions (blue circles) and actual positions (red triangles) are plotted in three-dimensional space. Dashed lines link each target-actual pair, highlighting minor deviations attributed to servo tolerance and mechanical backlash during joint movement. (b) Comparison of Target and Actual Place Coordinates in 3D Space. This illustrates the robotic arm's precision during object placement. Target locations (green circles) and corresponding actual coordinates (orange triangles) are plotted in 3D space. The dashed lines indicate the spatial deviation between target and actual endpoints, confirming the system's overall positional reliability under inverse kinematics control.

These visual plots confirm the effectiveness of the inverse kinematics module and servo control logic implemented via PWM on the ATmega328.

4.4 System Performance Evaluation

The robotic arm's performance was assessed based on:

i Positional Accuracy: Achieved sub-centimeter deviation in 4 out of 5 tests.
ii Responsiveness: Average movement time per task = 1.2–1.6 seconds.
iii Bluetooth Latency: Estimated at <100 ms per command, consistent with HC-05 module benchmarks [24].
iv Repeatability: Joint commands returned consistent positions within ±2° on repeat trials.

These metrics validate the arm's suitability for academic and low-precision industrial applications. Performance was slightly affected by load and arm extension length, which introduces additional torque demands, an observation consistent with findings in related Arduino-Raspberry Pi hybrid platforms [17].

5.0 Conclusion and Recommendations

5.1 Conclusion

This study successfully demonstrated the design and implementation of a 3-Degree-of-Freedom (3-DOF) robotic arm controlled via an ATmega328 microcontroller, with wireless command input through a Bluetooth-enabled Android interface. The project integrated mechanical design, embedded control logic, inverse kinematics algorithms, and human-machine interfacing into a cohesive system aimed at affordability, modularity, and educational value.

The experimental results confirmed that the robotic arm accurately executed object manipulation tasks within its defined workspace. Empirical data from both hardware trials and simulations indicated that the arm achieved high positioning precision, with end-effector deviations generally within ±2 cm of the target. The use of inverse kinematics for trajectory computation enabled effective joint coordination and motion control, while the integration of a mobile application provided a user-friendly and wireless means of command execution.

The simplicity of the system's architecture, combined with the flexibility of open-source development tools like Arduino and MIT App Inventor, reinforces its suitability for academic labs, STEM education, and early-stage automation prototyping. Furthermore, the study provides a replicable framework for other researchers or educators aiming to build low-cost robotic systems with functional autonomy.

Despite the successful outcomes, several limitations were identified. Mechanical backlash in servo joints, latency in Bluetooth transmission under interference, and lack of advanced feedback sensors constrained system accuracy under dynamic loads. These limitations present opportunities for refinement and future enhancement.

5.2 Recommendations

Based on the findings of this work, the following recommendations are proposed for future research, system improvement, and educational deployment:

i Advanced Control Algorithms: Future iterations of the robotic arm should integrate more sophisticated control strategies, such as Proportional-Integral-Derivative (PID) control or computed torque control, to improve stability, responsiveness, and error correction under varying loads.

ii Sensor Integration: Incorporating feedback mechanisms such as rotary encoders, inertial measurement units (IMUs), or force sensors would significantly enhance the precision and real-time adaptability of the arm. Sensor feedback can also enable closed-loop control and error detection.

iii Improved Mechanical Design: Reducing mechanical play through the use of higher-quality joints, gear trains, or 3D-printed linkages with tighter tolerances would mitigate backlash and improve repeatability. A redesign using brushless DC motors and load-bearing joints could expand the arm's payload capacity.

iv Expanded User Interface: A more intuitive human-machine interface (HMI) could be developed using graphical visualizations, gesture recognition, or voice control. Such features would enhance user engagement and usability, particularly in classroom demonstrations.

v Wireless Protocol Enhancement: While Bluetooth provided sufficient performance, migrating to higher-bandwidth, lower-latency communication protocols such as Wi-Fi or Zigbee could reduce latency and improve connectivity over longer distances or in interference-prone environments.

vi Battery-Powered Mobility: Introducing a battery-powered option would improve portability and reduce dependency on external power sources, making the system more suitable for mobile robotics applications and off-grid learning environments.

vii Educational Resource Development: To support adoption in academic settings, the development of step-by-step assembly guides, coding tutorials, and simulation exercises is recommended. These resources would lower the entry barrier for students and educators seeking hands-on experience in robotics.

viii Community Engagement and Open-Source Sharing: Publishing the complete design files, source code, and hardware schematics in open repositories (e.g., GitHub) would foster collaboration and iterative development within the academic and maker communities.

ix

By implementing these enhancements, the 3-DOF robotic arm platform can evolve into a more robust, intelligent, and scalable tool for both research and educational robotics. The foundational work presented in this study provides a strong baseline for continued exploration into accessible robotic systems and embedded control applications.

References

[1] Kruthika, K., Kumar, B. K., & Lakshminarayanan, S. (2016). Design and development of a robotic arm. *Proc. Int. Conf. on Circuits, Controls, Communications and Computing (I4C)*, pp. 1–4.

[2] Gautam, R., Gedam, A., Zade, A., & Mahawadiwar, A. (2017). Review on development of industrial robotic arm. *Int. Res. J. Eng. Technol.*, vol. 4, no. 3, pp. 429.

[3] Kadir, W. M. H. W., Samin, R. E., & Ibrahim, B. S. K. (2012). Internet controlled robotic arm. *Procedia Engineering*, vol. 41, pp. 1065–1071.

[4] Farman, M., Al-Shaibah, M., Aoraiath, Z., & Jarrar, F. (2018). Design of a three degrees of freedom robotic arm. *Int. J. Comput. Appl.*, vol. 179, no. 37, pp. 12–17.

[5] Smith, J. A., & Garcia, L. E. (2022). Development and control of a 3DOF robotic arm for educational purposes. *J. Robotics and Automation*, vol. 45, no. 2, pp. 112–127.

[6] Jiang, Z., Cao, X., Huang, X., Li, H., & Ceccarelli, M. (2022). Progress and development trend of space intelligent robot technology. *Space: Sci. & Technol.*

[7] Yunusa, M., Boya, S., Olatunde, A., Mushinwa, I., & CS, O. (2024). Four Arm Degree of Freedom 4DOF Robotic Arm. *Int. J. Eng. Process. and Safety Res.*

[8] Escudero, M. R., Hierro, C. M., & Pablo, A. P. D. M. (2013). Using Arduino to enhance computer programming courses in science and engineering. *EDULEARN13 Proc.*, pp. 5127–5133.

[9] Wagaa, N., Kallel, H., & Mellouli, N. (2023). Analytical and deep learning approaches for solving the inverse kinematic problem of a high-DOF robotic arm. *Eng. Appl. of Artif. Intell.*, vol. 123, 106301.

[10] Gutierrez, C. A. G., Reséndiz, J. R., Santibáñez, J. D. M., & Bobadilla, G. M. (2014). A model and simulation of a five-DOF robotic arm for mechatronic courses. *IEEE Latin America Trans.*, vol. 12, no. 2, pp. 78–86.

[11] Clark, S. J., & Smith, J. A. (2013). Sensor integration in a 3DOF robotic arm for precise manipulation. *Sensors and Actuators A: Physical*, vol. 21, no. 2, pp. 110–125.

[12] Patidar, V., & Tiwari, R. (2016). Survey of robotic arms and parameters. In *Proc. Int. Conf. on Computer Communication and Informatics (ICCCI)*, pp. 1–6.

[13] H. S. Kim and J. B. Song, "Low-cost robot arm with 3-DOF counterbalance mechanism," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 4183–4188, 2013.

[14] H. Mohammed Ali, Y. Hashim, and G. A. Al-Sakkal, "Design and implementation of Arduino-based robotic arm," *Int. J. Electrical and Computer Engineering*, vol. 12, no. 2, pp. 1411–1419, 2022.

[15] K. Vamshi, "Adaptive Intelligence in Warehouse Robotics: Efficient Pick-and-Place Robot for Dynamic Environments," *J. Electrical Systems*, vol. 20, no. 3, pp. 8215–8230, 2024.

[16] C. A. G. Gutierrez, J. R. Reséndiz, J. D. M. Santibáñez, and G. M. Bobadilla, "A model and simulation of a five-degree-of-freedom robotic arm for mechatronic courses," *IEEE Latin America Transactions*, vol. 12, no. 2, pp. 78–86, 2014.

[17] N. Tayebi and S. Patel, "Modeling and torque control of servo motors for robotic arm," *International Journal of Engineering Research and Technology (IJERT)*, vol. 6, no. 5, pp. 312–317, 2017.

[18] R. Sahraoui, "Selection of materials and design parameters for robotic arm joints," *Materials Today: Proceedings*, vol. 45, pp. 189–195, 2021.

[19] R. S. Khusainov and V. V. Zheleznov, "Optimization of robot arm structures using CAD and simulation tools," *Procedia CIRP*, vol. 76, pp. 411–416, 2018.

[20] A. Abhishek, S. Thomas, and R. Nair, "Design and implementation of a regulated power supply for microcontroller systems," *International Journal of Electronics and Electrical Engineering*, vol. 4, no. 2, pp. 55–59, 2016.

[21] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, Springer, 2011.

[22] H. Mohammed Ali, Y. Hashim, and G. A. Al-Sakkal, "Design and implementation of Arduino-based robotic arm," *Int. J. Electrical and Computer Engineering*, vol. 12, no. 2, pp. 1411–1419, 2022.