International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 14, Issue 5 [May. 2025] PP: 162-167

Technical Design and Control Implementation of a 6-DOF Robotic Arm for Industrial Applications within the MATLAB/SIMULINK Environment

Thi-Mai-Phuong Dao[1,*], Huy-Tuan Le[1], Van-Minh Pham[1], Kim-Thanh Tran[2], Xuan-Thu Le[1], Van-Thang Ngo[1]

[1]*Faculty of Automation, School of Electrical and Electronic Engineering (SEEE), Hanoi University of Industry, Hanoi, Vietnam.*

[2]*Faculty of Electrical Engineering, School of Electrical and Electronic Engineering (SEEE), Hanoi University of Industry, Hanoi, Vietnam.*

Abstract: *Robotic manipulators are extensively deployed in industrial manufacturing to mitigate manual labor, enhance throughput, and function in environments posing risks to human personnel. Notwithstanding their prevalent adoption, the research and design of robotic arms, particularly concerning performance optimization and precision enhancement, have not garnered substantial focus within numerous enterprises. This paper delineates a comprehensive research methodology for the design of a six-degree-of-freedom (6-DOF) robotic manipulator. The investigation commences with a rigorous analysis of the kinematic and dynamic challenges inherent to such systems. Subsequently, a 6-DOF robot model is developed and simulated utilizing SOLIDWORKS, complemented by the development of a control system architecture in MATLAB for real-time control and simulation. The proposed design endeavors to improve the accuracy and efficiency of robotic operations across diverse industrial applications.*

Keywords:*Robotic arm design, 6-DOF robotic arm, PID, fuzzy logic controller,simulation, MATLAB/Simulink*

Date of Submission: 10-05-2025 Date of acceptance: 20-05-2025

I. Introduction

Within the paradigm of the Fourth Industrial Revolution, the imperative for automation and augmented production efficiency has intensified considerably. Industrial robots, notably six-degree-of-freedom (6-DOF) robotic manipulators, constitute a pivotal element in advanced manufacturing paradigms owing to their aptitude for executing intricate tasks emulating human arm kinematics. These robotic systems facilitate high-precision and high-throughput operations, even within hazardous environments necessitating stringent safety protocols-conditions under which sustained human labor is challenging [1].

The 6-DOF robotic arms, characterized by their extensive range of articulated motion and multi-axial control capabilities, can effectively and precisely perform diverse tasks such as pick-and-place, assembly, packaging, painting, and welding [2]. Contemporary industries increasingly prioritize the integration of robots into assembly lines to elevate production rates and product quality [1-2]. The confluence of industrial robots with contemporary technologies, including CNC machining, cloud computing, and automated sensing systems, has catalyzed the evolution of flexible manufacturing systems, thereby propelling the prevailing trend toward "smart manufacturing" [1].

Extensive research has addressed this domain. For example, authors presented a methodology for developing a three-dimensional model and simulating inverse kinematics within the MATLAB environment, achieving a high degree of analytical accuracy [3]. Furthermore, the study [4] detailed the development of a remote control system for a 6-DOF robotic arm utilizing MATLAB, aimed at research and pedagogical applications. Moreover, recent years have seen a focus on optimizing control algorithms for 6-DOF manipulators. For instance, the study [5] introduced a novel optimization technique for motion planning that enhances the operational precision of robotic arms in complex workspaces.

For the efficacious development of industrial robot control systems, the simulation and validation of control algorithms in a computational environment prior to deployment on physical hardware is paramount. MATLAB/Simulink is recognized as a prevalent platform for robotic system simulation and design, attributed to its extensive suite of support libraries. Prior research has demonstrated that MATLAB and its associated toolboxes, such as the Robotics Toolbox and Simscape, facilitate comprehensive simulation of robotic systems, encompassing kinematic and dynamic analysis, sensor integration, machine vision, and support for the transition from simulation to physical implementation [5]. Employing the MATLAB environment enables developers to design and evaluate sophisticated control algorithms without the initial investment in expensive physical prototypes, thereby enhancing safety protocols and reducing development expenditures [5], [8]. Specifically, the Robotics Toolbox offers a diverse set of functions for the computation of kinematics, dynamics, and trajectory generation, which streamlines intuitive and efficient robot control programming and analysis [5], [6], [9].

The objective of this paper is to design a system and formulate a control strategy for a 6-DOF industrial robotic manipulator within the MATLAB platform to support contemporary manufacturing applications. The focus of this work is the construction of a complete kinematic model and the analytical solution of the inverse kinematics problem to achieve precise control over the end-effector's position and orientation. Prior investigations have established that solving the inverse kinematics equations for 6-DOF robots constitutes a fundamental prerequisite for directing the robot to desired spatial configurations [6]. Building upon this, control algorithms are developed and simulated in MATLAB/Simulink to assess the performance of the control system, contributing to the realization of more accurate and efficient automation solutions in industrial production [10].

The remainder of this paper is organized as follows. Section II introduces the materials and method for modeling a 6-DOFrobot. Section III presents the control method and discusses the simulation results implemented in MATLAB/Simulink. Section IV concludes the paper and suggests potential directions for future work.

II. Materials And Method

A six-degree-of-freedom (6-DOF) robot constitutes a complex electromechanical system capable of agile end-effector positioning and orientation within three-dimensional space. These six degrees of freedom typically correspond to three translational motions along the Cartesian axes (x, y, z) and three rotational motions about these axes, enabling the robot to access any attainable pose (position and orientation) within its operational workspace. Kinematic analysis of the robot is a fundamental aspect of its design and control. Forward kinematics pertains to the determination of the end-effector's pose based on known joint angle values. Conversely, inverse kinematics involves the computation of the requisite joint angle values to achieve a specific desired end-effector pose. The solution of the inverse kinematics problem is generally more intricate than that of forward kinematics and may yield multiple solutions or no solution. Both forward and inverse kinematics serve as foundational elements for trajectory planning, motion control, and the simulation of 6-DOF robots in industrial applications and research endeavors.

The block diagram to the studying 6-DOF robot model is illustrated in Figure 1. This diagram consists of five blocks as follows:

(1) Inputs: This block contains position matrix (Px, Py, Pz) and orientation matrix (Yaw, Pitch, Roll) as input reference values.
(2) Inverse Kinematics:When the desired values are received, this block calculates and outputs the joint angles (θ).
(3) Controller:Receives the joint angles and performs the corresponding control actions.
(4) 6-DOF robot model:This model receives control signals and simulates the robot's movement, then provides feedback on the actual joint states to the controller for use in the next control cycle. The dynamics model of a 6-DOF robot had been presented in [5, 8, 10].
(5) Outputs block:Receives the joint angles (θ_i) from the 6-DOF model.

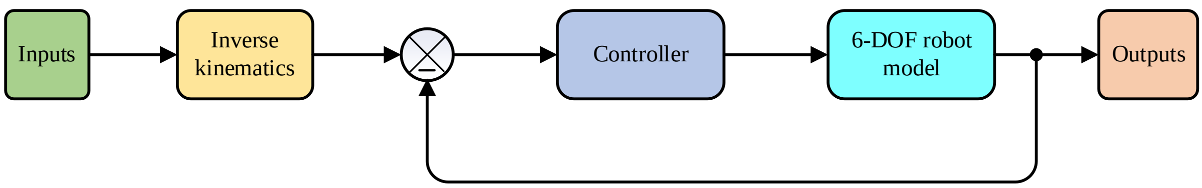

Fig. 1. Block diagram of the 6-DOF robotic arm system

III. Numerical Simulations On Matlab/Simulink

Based on the entired diagram shown in Fig. 1, it is straightforward to build the simulation diagram for the 6-DOF robot in MATLAB/SIMULINK environment as illustrated in Fig. 2. It is noted that the returned Theta Out values are calculated based on the position matrix [3x1] and the orientation matrix [3x1] of the forward kinematics, with results closely matching the input Theta values.Figure 3 shows a 6-DOF robotic arm modeled in SolidWorks and imported into MATLAB for motion simulation. This serves as the main actuator mechanism used to test control algorithms within the Simulink environmenthere.

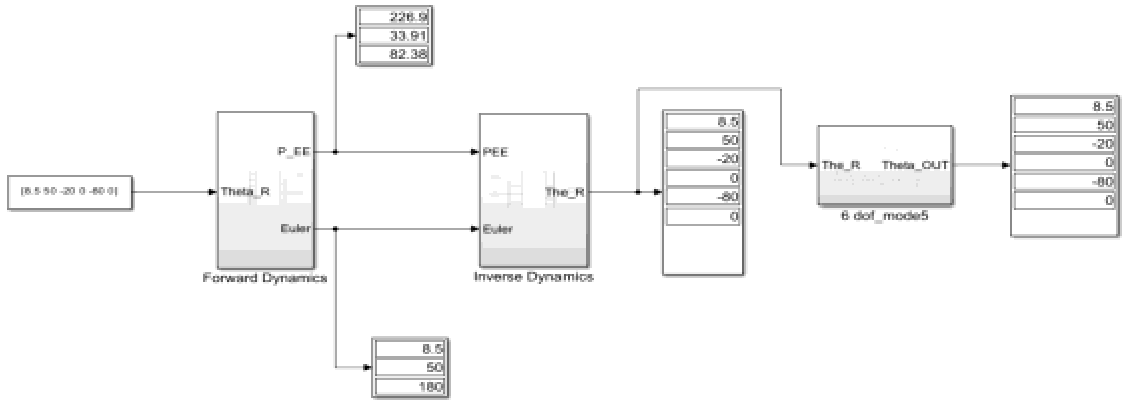

Fig. 2. Simulation diagram for the 6-DOF robot model

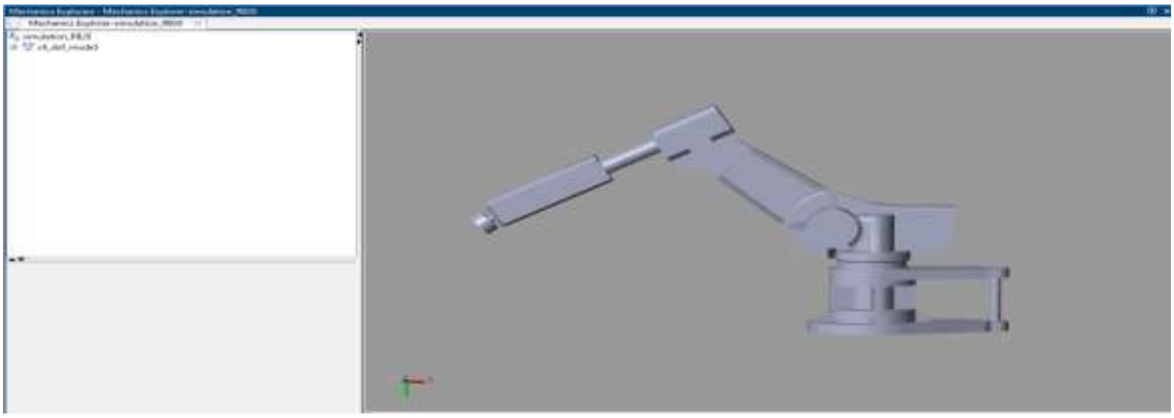

Fig. 4.Simulation of the robotic arm integrating MATLAB and SolidWorks

The application of Proportional-Integral-Derivative (PID) control to a 6-DOF robotic manipulator involves the independent or coupled regulation of each joint's angular position or velocity through the computation of control torques or forces based on the positional or velocity error, its integral, and its derivative. For each degree of freedom, the PID controller generates a control signal that is a linear combination of these three terms, aiming to minimize the tracking error between the desired and actual joint states. While conceptually straightforward, the effective tuning of the proportional (Kp), integral (Ki), and derivative (Kd) gains for a multi-joint system like a 6-DOF arm can be complex due to the inherent dynamic coupling and nonlinearities. Advanced techniques such as gain scheduling or decentralized PID control with gravity compensation are often employed to enhance performance and stability across the robot's operational workspace. The efficacy of the PID control scheme is typically assessed by evaluating the transient response characteristics, steady-state error, and robustness to disturbances in both simulation and experimental validation.Figure 5 describes a control system for the 6-DOF robotic arm built in MATLAB/SIMULINK under this study.

Besides the PID regulator, the implementation of a fuzzy logic controller (FLC) for a 6-DOF robotic manipulator presents a viable approach to managing the inherent nonlinearities and uncertainties within the system dynamics. By employing linguistic variables and fuzzy inference rules, the FLC can effectively map the error and its derivatives in joint space or Cartesian space to appropriate control actions, without requiring a precise mathematical model of the robot. This methodology offers robustness to parameter variations and external disturbances, enabling smoother and more adaptive trajectory tracking and regulation compared to conventional linear control techniques. The design of the FLC typically involves defining membership functions for the input and output variables, as well as formulating a rule base that encapsulates the desired control behavior, often derived from expert knowledge or experimental data. The performance of such a controller can be rigorously evaluated through simulation and experimentation, assessing metrics such as tracking accuracy, settling time, and overshoot in the operational workspace of the 6-DOF arm.

Upon the completion of the dynamic simulations, the resultant temporal evolutions of the joint angles for the 6-DOF robotic manipulator were recorded and analyzed. The obtained response characteristics suggest that both the implemented PID and FLC possess the capacity to yield acceptable control performance, as evidenced by the effective servoing of the robot's joints to their designated reference trajectories. By way of illustration, the fourth joint exhibited a high degree of fidelity in tracking its pre-specified angular profile. To facilitate a quantitative comparative assessment of the relative effectiveness of these two distinct control paradigms, a detailed analysis of the tracking error associated with the response of Joint 4 was undertaken. The findings of this comparative evaluation are visually presented in Fig. 7.

As clearly depicted in Fig. 7, the magnitude of the tracking error observed under the fuzzy logic control scheme is notably smaller than that exhibited by the PID controller specifically for the angular displacement θ_4. More precisely, the graphical representation corresponding to the FLC (indicated by the orange curve) demonstrates reduced oscillatory behavior and a diminished error amplitude when contrasted with the response under PID control (indicated by the blue curve). This empirical observation substantiates the assertion that the application of the fuzzy logic controller leads to enhanced accuracy and a reduction in undesirable oscillations within the control action at the robot's fourth joint.

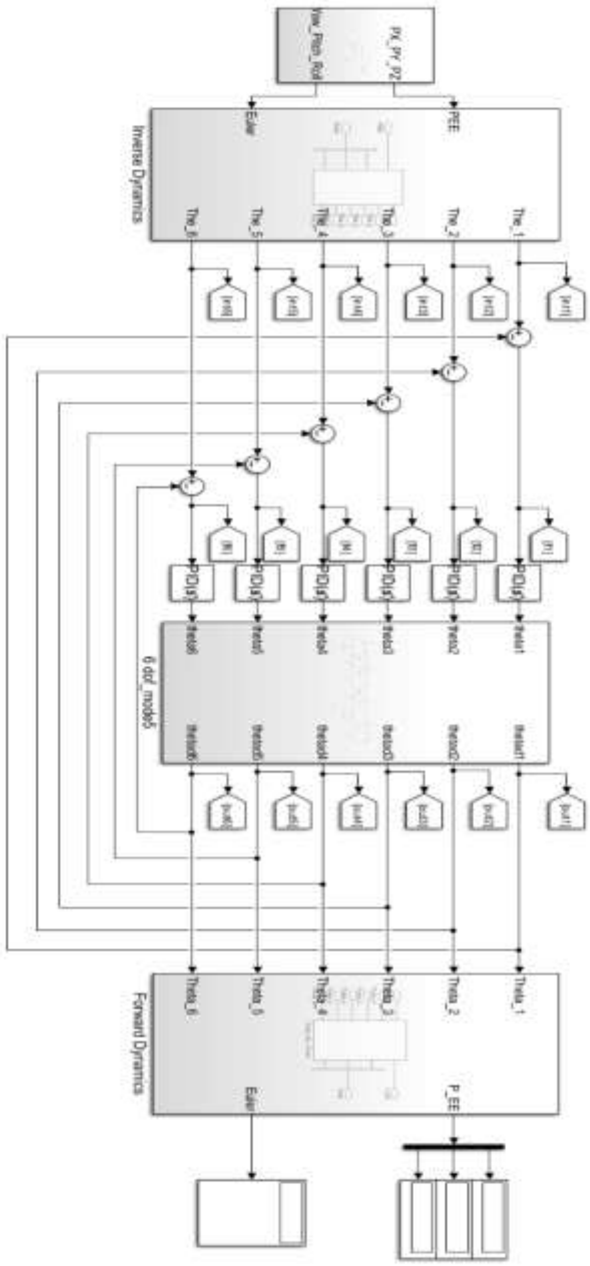

Fig. 5. Robot control simulation using the PID controller with trial-and-error method

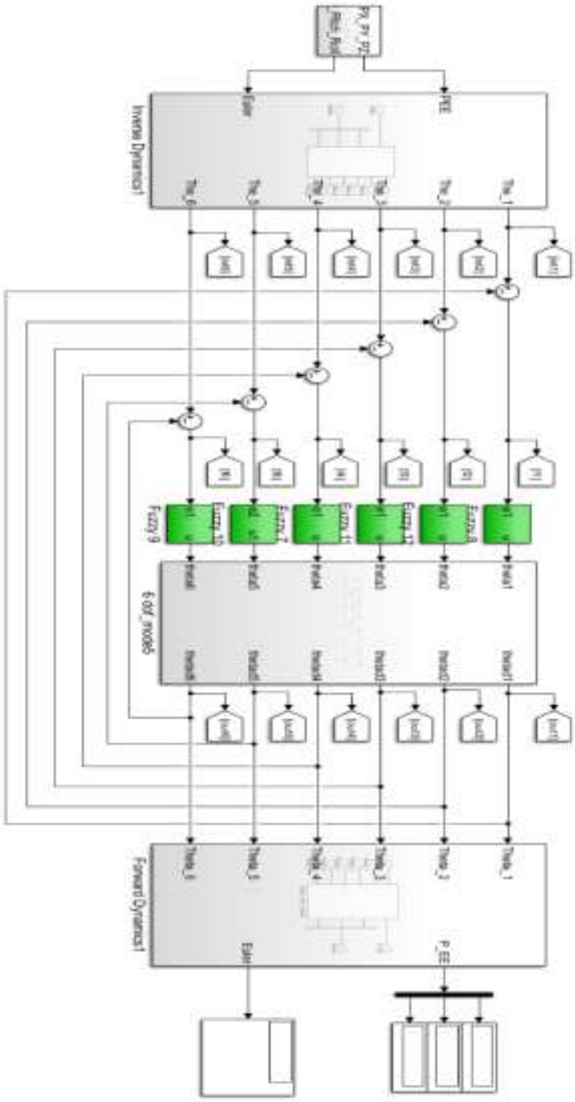

Fig. 6. Robot control simulation using the fuzzy logic controller

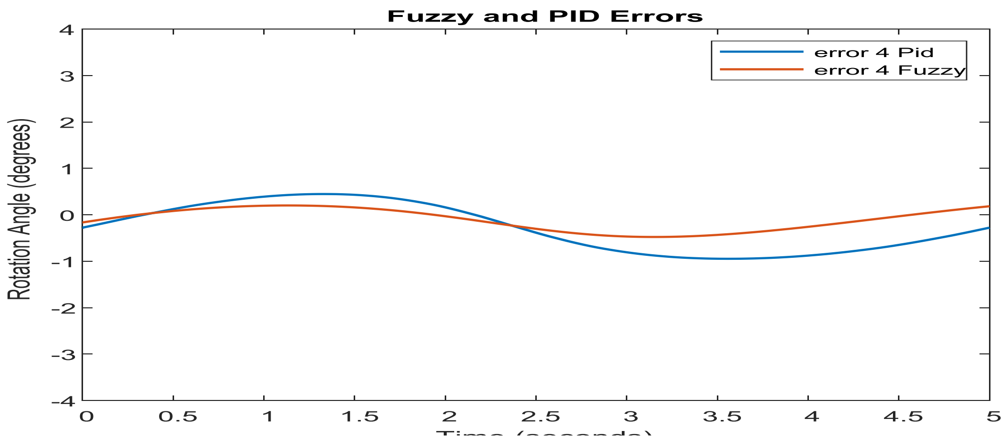

Fig. 7. Angular error of $\theta_4$for the PID and fuzzy logic controllers

IV. Conclusion and Future Work

This paper has presented a methodology for modeling a 6-DOF robotic arm utilizing the MATLAB/Simulink environment. Subsequently, both classical control methods (employing a PID controller) and an advanced control approach (utilizing a fuzzy logic controller) were designed. Simulation schematics of the controllers were implemented within the MATLAB/Simulink platform. The simulation results demonstrated that the joint angle trajectory tracking response of the fuzzy logic controller significantly outperformed that of the classical PID controller. Future research will focus on the industrial application of this 6-DOF robot, particularly in product classification and recognition problems leveraging artificial intelligence. This aligns directly with the current pervasive trends of the Fourth Industrial Revolution.

References

[1]. Y. Zhang và J. Li, "Do industrial robot applications upgrade the global value chain position of enterprises? Evidence from firm-level data," Journal of Asian Economics, vol. 85, 101604, 2025

[2]. S. Fujii và Q.-C. Pham (2021), "Realtime Trajectory Smoothing with Neural Nets", https://doi.org/10.48550/arXiv.2111.02165.

[3]. L. Žlajpah, B. Nemec và D. Omrcen, "MATLAB based robot control design environment for research and education," Proc. MATHMOD 2009, vol. 34–35, pp. 300–305, 2009.

[4]. G.ElAzab (2025), "6-DOF DLR Robot Simulation in Matlab/Simulink"

[5]. G. Gilca, "Kinematic modeling of 6 DOF robotic arm with MATLAB," Sci. Bull. "C. Brâncuşi" Univ. Târgu Jiu, Eng. Ser., no. 1, pp. 130–135, 2023.

[6]. V. Tinoco et al., "A review of advanced controller methodologies for robotic manipulators," Int. J. Dynamics and Control, vol. 13, Art. no. 36, 2025, doi: 10.1007/s40435-024-01533-1.

[7]. C. Chen, Z. Lv, S. Wu, K. Li, Z. Chen, and Z. Huang, "Design of control system for six-degree-of-freedom robotic arm," 2024 J. Phys.: Conf. Ser. 2724 012006, DOI 10.1088/1742-6596/2724/1/012006.

[8]. A. Iqbal and M. S. Khan, "Robotic arm design and control using MATLAB/Simulink," unpublished, 2023.

[9]. P. Mishra and A. R. Verma, "Forward and inverse kinematics of a 6-DOF robotic manipulator with MATLAB," Int. J. Adv. Technol. Eng. Explor., vol. 11, no. 1, pp. 25–30, Jan. 2024.

[10]. S. R. Kale, A. T. Waghmare, and D. P. Mahajan, "Trajectory optimization for 6-DOF robotic arm using WOA, GA, and PSO," Mater. Today: Proc., vol. 83, pp. 152–158, 2025.

[11]. I. Dynamics, "Forward and Inverse Kinematics: Explained," online at https://irisdynamics.com/articles/forward-and-inverse-kinematics#:~:text=The%20inverse%20kinematics%20translate%20the,Roll%2C%20Pitch%2C%20and%20Yaw.

[12]. M. Kumar, R. S. Jha, and V. R. Singh, "Image Processing Techniques for Object Detection in Robotics," Journal of Optical Engineering, vol. 62, no. 4, pp. 0984-0993, 2024.

[13]. S. A. Husi, "Hybrid Control Systems with Fuzzy Logic and PID for Improved Object Detection in Robotic Systems," International Journal of Control Engineering, vol. 30, no. 3, pp. 568-577, 2025.