

Fault Tolerant Control of Robot Manipulator with Actuator Effectiveness Adaptation

Anisa Ulya Darajat ^{1,*}, Swadexi Istiqphara ², Heriansyah ³, Mohammad Farhan Ferdous ⁴, Abu Saleh Musa Miah ⁵,
Uri Arta Ramadhani ⁶, Hari Maghfiroh ⁷

¹ Department of Electrical Engineering, Universitas Lampung, Bandar Lampung, Indonesia

^{2,3,6} Department of Electrical Engineering, Institut Teknologi Sumatera, Bandar Lampung, Indonesia

³ Institute of Telecooperation, Johannes Kepler University Linz, Altenberger Straße 69, Linz, Austria

⁴ Japan-Bangladesh Robotics and Advanced Technology Research Center (JBRATRC), Fukui, Japan

⁵ Department of Computer Science and Engineering, Rajshahi University (RU), Rajshahi, Bangladesh

⁷ Department of Electrical Engineering, Universitas Sebelas Maret, Surakarta, Indonesia

Email: ¹ anisa.ulya@eng.unila.ac.id, ² swadexi@el.itera.ac.id, ³ heri@el.itera.ac.id, ⁴ farhan.ferdous2019@gmail.com,

⁵ musa.cse@ru.ac.bd, ⁶ uri.ramadhani@el.itera.ac.id, ⁷ hari.maghfiroh@staff.uns.ac.id

*Corresponding Author

Abstract—This study discusses the implementation of fault-tolerant control (FTC) for a planar two-degree-of-freedom (2DoF) robot manipulator experiencing actuator loss of effectiveness. Several methods have been proposed, such as PID and MRAC; however, their accuracy still needs improvement. Meanwhile, FT-SMC offers high accuracy, but its methodological complexity results in longer execution time and reduced computational efficiency. The objective of this research is to develop a fault-tolerant control method that can maintain system performance under actuator degradation while achieving high tracking accuracy with improved computational efficiency. Simulations are performed with a two-link manipulator model with sinusoidal reference trajectories. An actuator fault is introduced at 4 s by reducing the actuator effectiveness to $[0.5, 0.7]^T$, meaning that the actuator capability decreases to 50% and 70% of its nominal performance, respectively. The simulation results show that the proposed FTC controller maintains good tracking performance after the fault occurs. In contrast, the controller without FTC experiences performance degradation characterized by phase lag and amplitude attenuation in the system response. Furthermore, the actuator effectiveness estimation mechanism demonstrates fast convergence after the fault occurs, with settling times of approximately 0.084 s and 0.238 s for the first and second joints, respectively. The steady-state MAEs are 0.0080 and 0.0395, equivalent to relative errors of 1.6% and 5.6%, respectively. Compared with other FTC methods, the proposed FTC controller also provides a balanced trade-off between tracking accuracy, robustness under fault conditions, and computational efficiency, making it suitable for real-time implementation.

Keywords—Fault-Tolerant Control; Robot Manipulator; Actuator Fault; Adaptive Control; Torque Control

I. INTRODUCTION

Robot manipulators are widely used across various industrial domains, including manufacturing, logistics, service applications, and inspection tasks. Their ability to pick and place objects efficiently makes them particularly valuable in environments where conventional conveyor systems are impractical. Moreover, their motion flexibility, especially when equipped with higher degrees of freedom, enables their use in complex applications such as welding and medical procedures [1].

To ensure precise and flexible operation, robot manipulators rely on actuators, typically DC motors, combined with position feedback sensors such as encoders or potentiometers. These actuators are placed at each joint, allowing coordinated motion and accurate control of the robot configuration. However, the robot actuators may lose their performance due to wear, heat, or impacts. This can lead to performance degradation and compromise the tracking controller's ability to accurately follow the desired trajectory, particularly when the controller is designed under the assumption of ideal actuator behavior [2].

Therefore, it is important to design the control strategy to maintain system performance even in the presence of actuator faults. Several control strategies can be employed to handle actuator faults, including adaptive and self-tuning control, which adjust controller parameters in response to system changes, as well as FTC, which is specifically designed to maintain stability and acceptable performance even in the presence of faults [3], [4].

Adaptive control approaches have been developed to adjust the parameters in the controller to take care of the faults in the actuators [5]-[7]. Sliding mode control approaches have also been developed to take care of uncertainties and disturbances in the system [2], [4], [8], [9]. New approaches have also been developed using neural networks, adaptive estimation, and learning approaches for fault handling in robot manipulators [10]-[13].

Advanced adaptive and robust control approaches have also been developed to take care of the reduced performance of the actuators in robot manipulators and to prevent system failure [14], [15]. Learning approaches such as reinforcement learning have also been developed to take care of faults in the actuators in robot manipulators [16], [17].

Despite recent advancements, significant challenges remain for FTC methods. Many existing approaches rely on highly accurate models, additional observers, or explicit fault detection mechanisms, which increase system complexity. Some robust control strategies require complex parameter tuning and may induce chattering when actual faults differ from assumed models [1], [3], [4]. Therefore, a simpler FTC

method is needed to ensure reliable performance under actuator degradation.

To address these challenges, this study proposes an FTC method for a planar 2DoF robot manipulator experiencing actuator loss of effectiveness. The proposed approach integrates a model-based computed-torque control framework with an adaptive mechanism that estimates actuator effectiveness and rescales the commanded torque. A non-FTC controller serves as a baseline for performance evaluation under actuator degradation.

II. PROBLEM FORMULATION

The problem formulation section is divided into two parts. The first part presents the robot manipulator model, which describes in detail the mathematical model used in the simulations. The second part focuses on the fault model, explaining how faults are injected into the system to emulate actuator degradation.

A. Robot Manipulator Model

The dynamics of a two-links (2-DoF) planar robot manipulator, which is shown in Fig. 1 are represented using the Lagrangian formulation as [6]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (1)$$

where $q = [q_1 \ q_2]^T$ denotes the joint angles; $M(q)$ is the inertia matrix, $C(q, \dot{q})$ the Coriolis and centrifugal matrix, $G(q)$ the gravity vector, and τ the actuator torque.

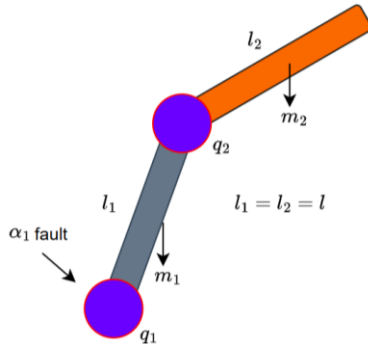


Fig. 1. 2 DoF planar robot manipulator

The link lengths, masses, inertias, distances to mass centers, and gravitational acceleration are defined in Table 1. The selected mass and link length values represent a lightweight two-link planar manipulator, which is commonly used in laboratory-scale robotic studies.

Table 1. Model Parameters

Parameter	Link 1	Link 2
Mass m (kg)	1.0	0.8
Length l (m)	0.5	0.4

In the case of a planar two-link manipulator, the standard dynamic formulation produces the inertia matrix, the Coriolis and centrifugal matrix, and the gravity vector.

$$M(q) = \begin{bmatrix} \frac{1}{3}m_1l^2 + \frac{4}{3}m_2l^2 + m_2l^2C_2 & \frac{1}{3}m_2l^2 + \frac{1}{2}m_2l^2C_2 \\ \frac{1}{3}m_2l^2 + \frac{1}{2}m_2l^2C_2 & \frac{1}{3}m_2l^2 \end{bmatrix} \quad (2)$$

$$C(q, \dot{q}) = \begin{bmatrix} -m_2S_2l^2\dot{q}_2 & -m_2S_2l^2(\dot{q}_1 + \dot{q}_2) \\ m_2S_2l^2\dot{q}_1 & 0 \end{bmatrix} \quad (3)$$

$$G(q) = \begin{bmatrix} \frac{1}{2}m_1glC_1 + \frac{1}{2}m_2glC_{12} + m_2glC_1 \\ \frac{1}{2}m_2glC_{12} \end{bmatrix} \quad (4)$$

where $C_1 = \cos(q_1)$, $C_2 = \cos(q_2)$, $C_{12} = \cos(q_1 + q_2)$, $S_2 = \sin(q_2)$. These expressions are directly used in the simulation model and controller design.

B. Fault Model

Actuator loss of effectiveness (LoE) denotes a reduction in input gain and serves as a standard abstraction for actuator degradation in manipulators. In this study, the plant torque is obtained by multiplying the torque generated by the controller with an unknown effectiveness factor. This formulation is expected to represent conditions such as actuator wear, partial loss of effectiveness, and the presence of disturbances, which may occur even in the absence of explicit mechanical or physical changes. The fault modeling is carried out based on this consideration and can be expressed as follows:

$$\tau = \text{diag}(\alpha)u + d \quad (5)$$

where u denotes the commanded torque generated by the controller. The unknown actuator fault for each joint is represented by $\alpha \in (0, 1]$. The term d represents a bounded disturbance arising from unmodeled dynamics and external perturbations. When $\alpha_i = 1$, the actuator operates under nominal conditions, while $\alpha_i < 1$ represents a proportional loss in the available torque. In the simulation, a sudden actuator fault is introduced at time t_f which is stated as follows:

$$\alpha_1(t) = \begin{cases} 1 & t < t_f \\ \alpha_{1,f} & t \geq t_f \end{cases} \quad (6)$$

The post-fault effectiveness values are defined as $\alpha_{1,f} = 0.5$ and $\alpha_{2,f} = 0.7$, representing 50% and 30% losses in actuator torque capability, respectively. These values were selected to represent different levels of actuator degradation commonly considered in fault-tolerant control studies, allowing the controller performance to be evaluated under asymmetric fault conditions. Such a fault scenario provides a more realistic assessment of the controller's ability to maintain tracking performance in the presence of partial actuator failures. The FTC scheme must address both actuator effectiveness loss and residual modeling uncertainties.

III. CONTROL DESIGN

This subsection presents the control design framework aimed at achieving accurate trajectory tracking of the robot manipulator in the presence of actuator degradation. The formulation focuses on the development of a model-based control law enhanced with robustness and adaptation mechanisms, which will be detailed in the following parts. The control system block diagram of the proposed FTC is shown in Fig. 2.

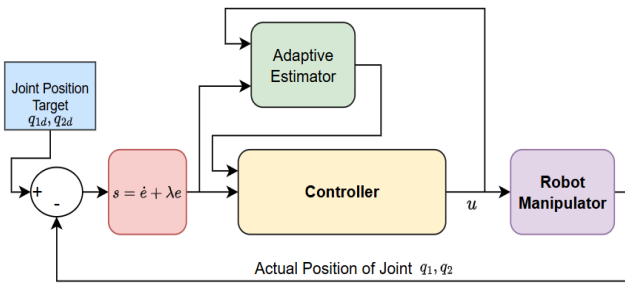


Fig. 2. FTC Controller

A. Tracking Error Definition

Let q_d denote the desired joint trajectory. The tracking error and its derivative are defined as follows, followed by the introduction of the filtered error signal:

$$e = q - q_d, \dot{e} = \dot{q} - \dot{q}_d \quad (7)$$

$$s = \dot{e} + \Lambda e \quad (8)$$

where Λ denotes a positive definite gain matrix, and the term \dot{e} denotes the time derivative of the tracking error. The filtered error s is defined such that $s \rightarrow 0$, the tracking error $e \rightarrow 0$, with the convergence rate determined by Λ . The primary control objective is to drive s toward zero despite effectiveness loss and bounded disturbances. The time derivative of s The basis for establishing the closed-loop error dynamics is stated as follows:

$$\dot{s} = \ddot{q} - \ddot{q}_d + \Lambda \dot{e} \quad (8)$$

where $\ddot{q} \in \mathbb{R}^n$ denotes the actual joint acceleration vector, $\ddot{q}_d \in \mathbb{R}^n$ represents the second time derivative of the desired joint trajectory.

B. Computed-Torque Control with Robust Compensation

To achieve accurate trajectory tracking for the nonlinear robotic system, a model-based control framework is adopted using the computed-torque control (CTC) approach. This method enables the cancellation of known nonlinear dynamics, thereby transforming the system into an equivalent linear error dynamic. To facilitate the controller design, a reference acceleration is first defined, and subsequently, the desired control torque is constructed based on the nominal model augmented with the robust compensation term and is expressed as follows,

$$v = \ddot{q}_d - \Lambda \dot{e} \quad (10)$$

$$\tau_{des} = M(q)v + C(q, \dot{q})\dot{q} + G(q) - K_s s \quad (11)$$

where K_s is a positive definite gain matrix that introduces damping into the filtered error dynamics and enhances robustness against disturbances.

This formulation removes the nominal nonlinear dynamics of the manipulator and incorporates a stabilizing term proportional to s . Under ideal conditions, where the actuators operate as specified ($\alpha = 1$), disturbances are absent ($d = 0$), and the commanded torque equals the

desired torque ($u = \tau_{des}$). By substituting the proposed control law into the manipulator dynamics under ideal conditions (i.e., $\alpha = 1$, $d = 0$, and $u = \tau_{des}$), the resulting closed-loop error dynamics can be expressed as,

$$M(q)\dot{s} + K_s s = 0 \quad (12)$$

This equation indicates exponential convergence of s , which consequently ensures convergence of the tracking error e . The inclusion of the robust term $K_s s$ also improves disturbance rejection and mitigates the influence of modeling inaccuracies.

C. Adaptation for Actuator Effectiveness

To compensate for actuator degradation, the FTC controller introduces an estimate $\hat{\alpha}$ of the actuator effectiveness. The control input is adjusted, and the parameter estimate is updated through the adaptation rule, which is expressed in the following equations:

$$u = \text{diag}\left(\frac{1}{\hat{\alpha}}\right) \tau_{des} \quad (13)$$

$$\dot{\hat{\alpha}} = \Gamma(u \circ s) \quad (14)$$

where Γ is a diagonal adaptation gain matrix and \circ denotes element-wise multiplication. This adaptation adjusts the estimated effectiveness based on the tracking error. If the error shows insufficient actuation, the estimate increases, scaling up the torque. If the error changes direction, the update reduces the estimate to avoid overcompensation. To secure stability and feasibility, the estimate is kept within the interval using a projection mapping.

D. Closed-Loop Error Dynamics

If actuator faults and disturbances are present, the actual plant torque can be expressed, and substituting the FTC control input yields the resulting error dynamics, which is can be expressed as follows,

$$\tau = \text{diag}(\alpha)u + d \quad (15)$$

$$M(q)\dot{s} + K_s s = \left(I - \text{diag}\left(\frac{\alpha}{\hat{\alpha}}\right)\right) \tau_{des} + d \quad (16)$$

This relation highlights the importance of the adaptation mechanism. As the estimate $\hat{\alpha}$ approaches the true actuator effectiveness α , the multiplicative mismatch term diminishes, and the nominal closed-loop dynamics are recovered. During transient phases, the robust term $K_s s$ acts to suppress disturbances and parameter mismatch, ensuring bounded tracking errors and improved recovery performance.

E. Baseline Controller (Without FTC)

For comparison purposes, a baseline controller without fault-tolerant capability is also considered. In this case, the control input is simply stated as:

$$u = \tau_{des}, \dot{\hat{\alpha}} = 0 \quad (17)$$

Under this assumption, the controller operates as if the actuators are fully effective. Consequently, any reduction in actuator effectiveness directly decreases the available torque and weakens the nonlinear cancellation provided by the computed-torque control law. This limitation motivates the incorporation of the proposed FTC mechanism.

IV. RESULT AND DISCUSSION

In this study, the experiments were carried out through computer simulations. To evaluate the controller's capability in responding to faults, an actuator loss-of-effectiveness (LOE) fault is introduced at $t_f = 4$ s (where t_f denotes the fault time). The actuator effectiveness values are set to $\alpha = [0.5, 0.7]^T$ to represent partial actuator degradation, meaning that the actuators can only generate 50% and 70% of their nominal torque capability, respectively.

Fig. 3 shows the tracking performance of a 2DoF of robot manipulator in terms of joint positions and joint velocities for two control schemes, namely the proposed fault tolerant control (FTC) controller and a controller without FTC used as a comparison.

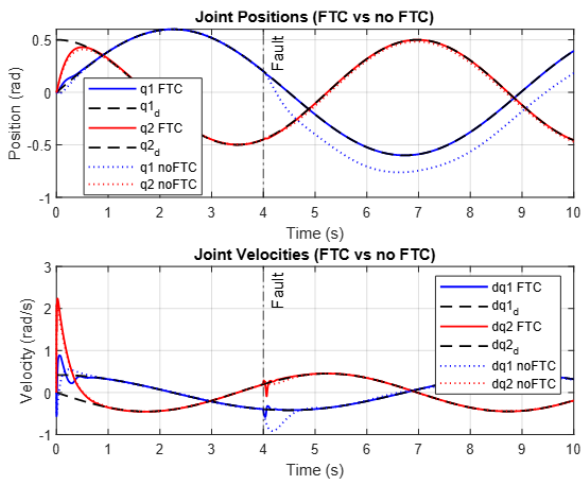


Fig. 3. Performance comparison on tracking the desired position

The upper graph in Fig. 3 illustrates the responses of the joint positions q_1 and q_2 , while the lower graph in Fig. 3 shows the joint velocity responses \dot{q}_1 and \dot{q}_2 . In this study, before the fault occurs at $t_f = 4$ s, both controllers exhibit good tracking performance. The joint positions closely follow the reference trajectories, and the velocity responses show relatively small tracking errors. This occurs because, under the initial condition, the actuators operate normally, allowing both controllers to effectively compensate for the system dynamics.

However, once the actuator loses effectiveness at $t_f = 4$ s, the controller without FTC begins to show degraded tracking performance. The joint positions deviate from the reference trajectories, accompanied by phase lag and reduced amplitude. Furthermore, the velocity responses present larger transient peaks.

Fig. 4 illustrates the evolution of the estimated actuator effectiveness parameters $\hat{\alpha}_1$ and $\hat{\alpha}_2$, providing insight into the behavior of the adaptive mechanism. Prior to the fault, both estimates remain close to the nominal value, indicating that the controller does not introduce unnecessary adaptation in

the absence of degradation. Once the fault is introduced at $t_f = 4$ s, a clear transient response is observed, reflecting the adaptation process triggered by the change in actuator effectiveness.

Following this transient phase, the estimates gradually converge toward the actual values, approximately $\alpha_1 f = 0.5$ and $\alpha_2 f = 0.7$. Although a small estimation error may still be present, the convergence trend confirms that the adaptive mechanism is able to capture the loss of effectiveness with reasonable accuracy. More importantly, the relatively fast convergence allows the controller to adjust the control input in a timely manner, thereby limiting performance degradation and maintaining acceptable tracking behavior despite the actuator fault.

To evaluate the estimator's performance, Table 2 summarizes the adaptive estimation metrics used in this study. The settling time is defined as the earliest time after the fault occurs when the estimation error $|\hat{\alpha}_i - \alpha_i|$ stays within a tolerance band of 0.1. Based on this definition, the estimate $\hat{\alpha}_1$ reaches a steady condition at approximately 0.084 s after the fault, while $\hat{\alpha}_2$ settles at around 0.238 s.

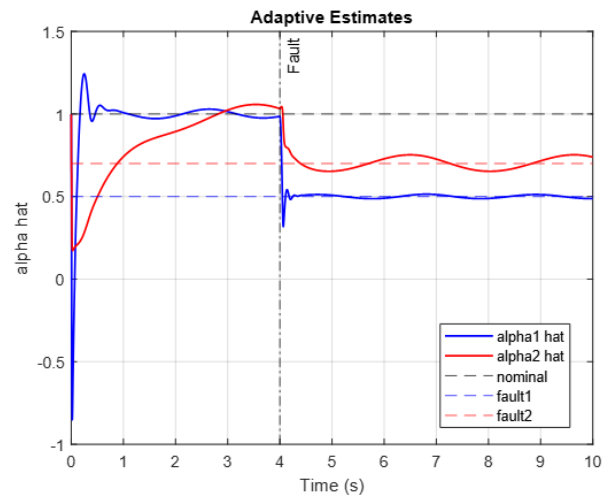


Fig. 4. Estimated Actuator Effectiveness

Table 2. Adaptive Estimation Metrics

Estimated	Settling Time After Fault (seconds)	Mean Absolute Error (MAE) (8–10seconds)	Relative MAE (%)
$\hat{\alpha}_1$	0.084	0.0080	1.6
$\hat{\alpha}_2$	0.238	0.0395	5.6

The steady-state estimation accuracy is then evaluated using the mean absolute error (MAE), which is the average absolute difference between the estimated and actual actuator effectiveness over the final 2 seconds of the simulation (8–10 seconds). The results show that the MAE for $\hat{\alpha}_1$ is 0.0080, while for $\hat{\alpha}_2$ it is 0.0395. These MAE values correspond to relative estimation errors of 1.6% and 5.6%, respectively. With these estimates available, the controller can appropriately rescale the control input, effectively compensating for actuator degradation.

The additional test compares the proposed method with the method from the paper using post-fault tracking metrics and execution time. The tracking metrics are computed over $t \geq t_f$ and the performance comparison of the various FTC control methods is shown in Table 3.

From Table 3, it can be seen that FTSMC-FC achieves the lowest tracking error among all compared methods because its fast terminal sliding mode structure drives the tracking error toward the sliding surface with rapid convergence, even under disturbances and actuator faults. In addition, this method uses a scheduled post-fault compensation factor, where the actuator effectiveness after the fault is assumed to be known and is directly incorporated into the control input correction. This combination allows the controller to apply a more aggressive and accurate control torque according to the post-fault plant condition, resulting in the lowest RMSE and peak error. However, this performance comes at the cost of higher computational complexity, leading to the longest execution time. The FTSMC-FC control law involves nonlinear operations such as terminal power terms, saturation functions, and fault compensation at each control update. As a result, the total simulation time becomes longer compared to other controllers with simpler structures.

The proposed method gives the best result among the PID+FC and MRAC-PID, with lower position RMSE and IAE than MRAC-PID and much smaller errors than the no-FTC baseline. The execution time of the proposed method is higher than that of PID+FC and MRAC-PID because the inclusion of adaptive actuator effectiveness estimation and robust compensation introduces additional numerical computations during each control update. However, its computational cost remains lower than FTSMC-FC, which involves more complex nonlinear control operations.

The comparative performance of the FTC methods can be seen in Fig. 5. Based on the results, particularly in the Joint 1 comparative tracking plot, most methods show relatively similar tracking performance. However, MRAC-PID appears to be more affected by the fault, as indicated by a more significant deviation from the reference trajectory. In contrast, the other methods remain relatively robust under fault conditions, showing only minor deviations after the fault occurs.

Table 3. Performance Comparison of Various FTC Methods

Method	RMSE (q)	RMSE (\dot{q})	Peak (q)	SS MAE (q)	Time (s)
Proposed FTC	0.0019	0.0289	0.0095	0.0004	0.4222
PID + FC [18]	0.0048	0.0033	0.0104	0.0030	0.3742
MRAC-PID [19]	0.0062	0.0315	0.0365	0.0017	0.2384
FTSMC-FC [20]	0.0003	0.0007	0.0004	0.0002	1.6631
No FTC	0.1255	0.0797	0.2114	0.1077	0.1497

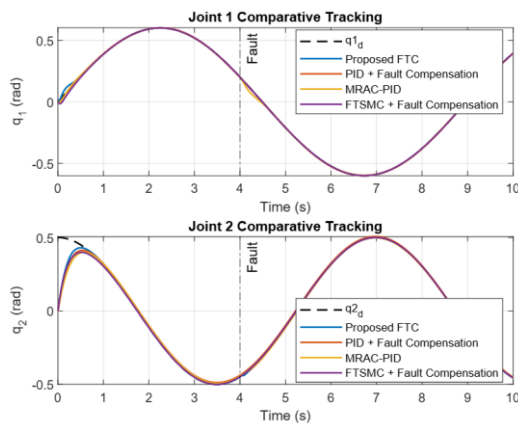


Fig. 5. Performance comparison on tracking the desired position

V. CONCLUSION

This study proposes an FTC method for a planar 2DoF robot manipulator experiencing actuator loss of effectiveness. The proposed approach enables the system to maintain trajectory-tracking performance despite actuator degradation. Simulation results show that the proposed FTC controller maintains tracking performance even when actuator faults occur. When a fault is introduced at $t_f = 4s$ with actuator effectiveness values reduced to $\alpha = [0.5, 0.7]^T$, the controller without FTC experiences performance degradation characterized by phase lag and amplitude attenuation. In contrast, the FTC controller can maintain stable position and velocity tracking by appropriately rescaling the control torque.

Furthermore, the actuator effectiveness estimates reach steady conditions approximately 0.084s and 0.238s after the fault occurs, with MAE values of 0.0080 and 0.0395, corresponding to relative errors of 1.6% and 5.6%, respectively. These results indicate that the proposed FTC method is able to compensate for actuator degradation and maintain reliable trajectory tracking performance. Future work will focus on extending the approach to more complex robotic systems.

In comparison with other FTC approaches, the proposed method provides a balanced performance between tracking accuracy, robustness, and computational efficiency. Although FTSMC-FC achieves the lowest tracking error, it requires significantly higher computational effort. On the other hand, PID+FC and MRAC-PID offer faster execution times but show lower robustness or reduced tracking performance under fault conditions. The proposed method maintains competitive tracking accuracy while preserving moderate computational cost, making it more practical for real-time fault-tolerant control applications.

Overall, the results confirm that the proposed method can improve the robustness and fault recovery capability of the manipulator under partial actuator failures. However, this study is currently limited to simulation-based validation under predefined fault scenarios and model uncertainties. Real-world implementation may introduce additional challenges such as sensor noise, actuator nonlinearities, and computational constraints. Therefore, future work will focus on experimental validation on a physical robotic platform and the extension of the proposed FTC framework to higher-degree-of-freedom manipulators operating under more complex fault conditions.

REFERENCES

- [1] A. Milecki and P. Nowak, "Review of Fault-Tolerant Control Systems Used in Robotic Manipulators," *Applied Sciences (Switzerland)*, vol. 13, no. 4, p. 2675, 2023, <https://doi.org/10.3390/app13042675>.
- [2] Z. Khan, A. Nasir, and S. Mekid, "Fault-tolerant control strategies for industrial robots: state of the art and future perspective on AI-based fault management," *Artificial Intelligence Review*, vol. 58, no. 11, p. 362, Aug. 2025, <https://doi.org/10.1007/s10462-025-11327-2>.
- [3] M. Baioumy, C. Pezzato, R. Ferrari, C. H. Corbato, and N. Hawes, "Fault-tolerant control of robot manipulators with sensory faults using unbiased active inference," in *2021 European Control Conference, ECC 2021*, pp. 1119–1125, 2021, <https://doi.org/10.23919/ECC54610.2021.9654913>.
- [4] G. Zhang, H. Liu, Z. Qin, G. V. Moiseev, and J. Huo, "Research on Self-Recovery Control Algorithm of Quadruped Robot Fall Based on Reinforcement Learning," *Actuators*, vol. 12, no. 3, p. 110, 2023, <https://doi.org/10.3390/act12030110>.

- [5] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, 2020, <https://doi.org/10.1126/SCIROBOTICS.ABC5986>.
- [6] Z. Dachang, H. Pengcheng, D. Baolin, and Z. Puchen, "Adaptive nonsingular terminal sliding mode control of robot manipulator based on contour error compensation," *Scientific Reports*, vol. 13, no. 1, 2023, <https://doi.org/10.1038/s41598-023-27633-0>.
- [7] Y. Wang and M. Wang, "Adaptive Fault-Tolerant Sliding Mode Control Design for Robotic Manipulators with Uncertainties and Actuator Failures," *Symmetry*, vol. 17, no. 9, p. 1547, 2025, <https://doi.org/10.3390/sym17091547>.
- [8] Q. D. Le and H. J. Kang, "Implementation of fault-tolerant control for a robot manipulator based on synchronous sliding mode control," *Applied Sciences (Switzerland)*, vol. 10, no. 7, p. 2534, 2020, <https://doi.org/10.3390/app10072534>.
- [9] J. Pan, L. Qu, and K. Peng, "Fault-Tolerant Control of Multi-Joint Robot Based on Fractional-Order Sliding Mode," *Applied Sciences (Switzerland)*, vol. 12, no. 23, p. 11908, 2022, <https://doi.org/10.3390/app122311908>.
- [10] S. M. Ebrahimi, F. Norouzi, H. Dastres, R. Faieghi, M. Naderi, and M. Malekzadeh, "Sensor fault detection and compensation with performance prescription for robotic manipulators," *Journal of the Franklin Institute*, vol. 361, no. 7, p. 106742, 2024, <https://doi.org/10.1016/j.jfranklin.2024.106742>.
- [11] Z. Feng and G. Hu, "Formation Tracking of Multiagent Systems with Time-Varying Actuator Faults and Its Application to Task-Space Cooperative Tracking of Manipulators," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 3, pp. 1156–1168, 2023, <https://doi.org/10.1109/TNNLS.2021.3104987>.
- [12] X. Zhang, Z. Yang, H. Liu, and X. Huang, "Optimal Sliding Mode Fault-Tolerant Control for Multiple Robotic Manipulators via Critic-Only Dynamic Programming," *Sensors*, vol. 25, no. 17, p. 5410, 2025, <https://doi.org/10.3390/s25175410>.
- [13] Q. D. Le and E. Yang, "Adaptive Fault-Tolerant Tracking Control for Multi-Joint Robot Manipulators via Neural Network-Based Synchronization," *Sensors*, vol. 24, no. 21, p. 6837, 2024, <https://doi.org/10.3390/s24216837>.
- [14] Y. Chen, J. Ding, Y. Chen, and D. Yan, "Nonlinear Robust Adaptive Control of Universal Manipulators Based on Desired Trajectory," *Applied Sciences (Switzerland)*, vol. 14, no. 5, p. 2219, 2024, <https://doi.org/10.3390/app14052219>.
- [15] I. Ali, M. Hassan, Z. Bano, E. Jawo, and M. M. A. Almazah, "Robust control of robot manipulator dynamics with two stages algorithm of optimal and integral sliding mode approaches," *Scientific Reports*, vol. 15, no. 1, p. 36585, 2025, <https://doi.org/10.1038/s41598-025-20478-9>.
- [16] F. Zhang, W. Wu, R. Song, and C. Wang, "Dynamic learning-based fault tolerant control for robotic manipulators with actuator faults," *Journal of the Franklin Institute*, vol. 360, no. 2, pp. 862–886, 2023, <https://doi.org/10.1016/j.jfranklin.2022.11.044>.
- [17] Z. Anjum, Z. Sun, S. Ahmed, and A. T. Azar, "Adaptive fixed-time fault-tolerant trajectory tracking control for disturbed robotic manipulator," *Plos One*, vol. 20, no. 6, 2025, <https://doi.org/10.1371/journal.pone.0323346>.
- [18] S. F. A. Latip, A. R. Husain, M. N. Ahmad, and Z. Mohamed, "Fault tolerant control for sensor fault of a single-link flexible manipulator system," *Jurnal Teknologi*, vol. 78, no. 6–13, pp. 59–66, 2016, <https://doi.org/10.11113/jt.v78.9273>.
- [19] Y. Ting, S. Tosunoglu, and D. Tesar, "Control structure for fault-tolerant operation of robotic manipulators," in *Proceedings - IEEE International Conference on Robotics and Automation*, 1993, pp. 684–690, <https://doi.org/10.1109/robot.1993.291826>.
- [20] M. Van, S. S. Ge, and H. Ren, "Finite Time Fault Tolerant Control for Robot Manipulators Using Time Delay Estimation and Continuous Nonsingular Fast Terminal Sliding Mode Control," *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1681–1693, 2017, <https://doi.org/10.1109/TCYB.2016.2555307>.