engineering proceedings

MDPI

Proceeding Paper

Comparative Evaluation of Lyapunov-Based Kinematic Controllers for Trajectory Tracking of Redundant Industrial Manipulators †

Mauricio Xavier López [1,2,*], Jesús Guamán-Molina [2,*], Marco Antonio Vasquez [3] and Luis Antonio Flores Asimbaya [2]

[1] Faculty of Engineering, Universidad Católica Andrés Bello, Caracas 1020, Venezuela
[2] Faculty of Systems, Electronics and Industrial, Universidad Técnica de Ambato, Ambato 180104, Ecuador; la.flores@uta.edu.ec
[3] Higher Technology Program in Mechatronics, Instituto Superior Tecnológico Martha Bucaram de Roldós, Nueva Loja 210150, Ecuador; marcovasquez@istmbr.edu.ec
* Correspondence: mxlopez.25@est.ucab.edu.ve (M.X.L.); jguaman0585@uta.edu.ec (J.G.-M.)
† Presented at the XXXIV Conference on Electrical and Electronic Engineering (JIEE 2026), Quito, Ecuador, 26–30 October 2026.

Abstract

Industrial robotic manipulators require control strategies that achieve accurate trajectory tracking without generating excessive joint-velocity commands. However, proprietary control architectures and the limited flexibility of physical platforms can restrict the preliminary evaluation of alternative controllers. This study develops a MATLAB R2025B-based virtual model of a Mitsubishi MELFA RV-2FRV robotic manipulator, represented through a reduced four-degree-of-freedom configuration, to compare kinematic control strategies based on a Lyapunov design criterion. The robot geometry was reconstructed from CAD components exported in STL format, while forward kinematics, the Jacobian pseudoinverse, position regulation, trajectory tracking, null-space control, and nonlinear velocity attenuation were implemented. The controllers were evaluated using circular, figure-eight, heart-shaped, and point-defined reference paths. Standardized tests produced nearly identical circular-path RMSE for Controllers 1 and 2 (0.00953 and 0.00949 m), while the null-space controller increased peak joint velocity from 0.681 to 1.330 rad/s. On the common heart-shaped path, nonlinear attenuation slightly increased RMSE to 0.03089 m but reduced peak joint velocity to 8.171 rad/s, confirming that tracking accuracy and joint-space demand should be evaluated together.

Keywords: industrial robotics; trajectory tracking; kinematic control; null-space control; velocity attenuation; MATLAB simulation

Check for updates

Academic Editor: Jackeline Abad

Published: 2 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Industrial robotics is a central technology for flexible manufacturing, assembly, and inspection within Industry 4.0 [1,2]. Before physical implementation, virtual models allow control laws, workspace restrictions, and demanding trajectories to be evaluated without exposing equipment or operators to unnecessary risk [3,4]. MATLAB-based environments are particularly useful for manipulator modeling because they support forward and inverse kinematics, Jacobian calculations, trajectory generation, and controller assessment [5].

A recurring challenge in manipulator control is that a reduction in Cartesian tracking error may require larger joint velocities. This compromise becomes relevant near path

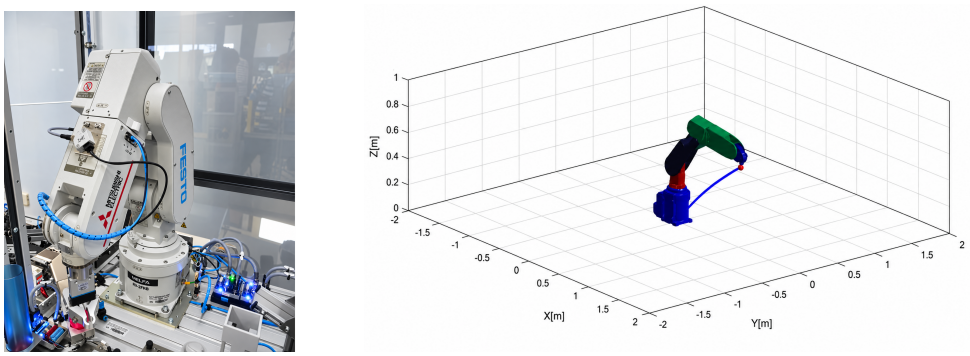
discontinuities and poorly conditioned configurations, where pseudoinverse-based solutions can amplify velocity commands. Lyapunov-based formulations provide a systematic framework for analyzing the convergence of tracking errors, while null-space terms allow a secondary joint-space objective to be introduced without modifying the primary Cartesian task [6–9]. Recent studies continue to compare controller structures for robotic manipulators, emphasizing the need to evaluate stability, response speed, and tracking accuracy jointly rather than as isolated indicators.

This proceeding paper presents a concise comparative evaluation of three kinematic control formulations for a reduced four-degree-of-freedom (4-DOF) representation of a Mitsubishi MELFA RV-2FRV manipulator: a primary Cartesian controller, a controller incorporating a null-space secondary task, and a nonlinear velocity-attenuation controller. Rather than proposing a new control-theory formulation, the contribution of this work is a comparative simulation study that quantifies the trade-off among Cartesian tracking accuracy, joint-velocity demand, trajectory smoothness, and Jacobian conditioning under common numerical conditions.

2. Materials and Methods

2.1. Virtual Robot and Kinematic Model

The physical six-axis Mitsubishi MELFA RV-2FRV was used as the geometric reference for the reduced 4-DOF model. The reduced coordinates retain base yaw and three serial pitch motions (physical J1, J2, J3, and wrist-pitch J5), while wrist-twist J4 and wrist-roll J6 were omitted because only end-effector position is controlled. The base frame has vertical z; q_1 rotates about z and q_2–q_4 rotate about parallel pitch axes. CAD parts were exported as STL meshes and assembled in MATLAB (Figure 1).



(**a**) (**b**)

Figure 1. Robot used as the modeling reference: (**a**) physical Mitsubishi MELFA platform and (**b**) reduced 4-DOF virtual model assembled in MATLAB.

The geometric parameters retained in the reduced model are summarized in Table 1. Cartesian coordinates were expressed in meters and joint variables in radians. A graphical scale factor of 1.5 was used only to visualize the STL meshes; the kinematic equations used the nominal dimensions. The analytical forward kinematics were cross-checked against the homogeneous-transform chain used to position the virtual links for five representative joint configurations; the maximum Cartesian discrepancy was 2.8×10^{-17} m, i.e., numerical round-off.

For the joint vector $\mathbf{q} = [q_1\ q_2\ q_3\ q_4]^T$, the end-effector position $\mathbf{h} = [h_x\ h_y\ h_z]^T$ was calculated [10].

$$h_x = \cos q_1[l_1 \cos q_2 + l_2 \cos(q_2 + q_3) + l_3 \cos(q_2 + q_3 + q_4)], \quad (1)$$

$$h_y = \sin q_1[l_1 \cos q_2 + l_2 \cos(q_2 + q_3) + l_3 \cos(q_2 + q_3 + q_4)], \quad (2)$$

https://doi.org/10.3390/engproc2026158022

$$h_z = b + l_1 \sin q_2 + l_2 \sin(q_2 + q_3) + l_3 \sin(q_2 + q_3 + q_4). \tag{3}$$

The relationship between Cartesian and joint velocities was

$$\dot{\mathbf{h}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \tag{4}$$

where $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{3\times4}$ is the geometric Jacobian. Because the model is redundant for a three-dimensional position task, the Moore–Penrose pseudoinverse $\mathbf{J}^{\dagger}$ was used to compute joint-velocity references [10].

Table 1. Geometric parameters of the reduced kinematic model.

Component	Symbol	Nominal Value (m)
Base height	b	0.295
First-link length	l_1	0.234
Second-link length	l_2	0.284
Third-link length	l_3	0.060

2.2. Control Laws and Stability Condition

The Cartesian tracking error was defined as $\mathbf{e} = \mathbf{h}_d - \mathbf{h}$, where $\mathbf{h}_d$ is the desired end-effector position. Under the assumption of ideal internal joint-velocity tracking, Controller 1 was defined by

$$\dot{\mathbf{q}}_{\text{ref},1} = \mathbf{J}^{\dagger}(\dot{\mathbf{h}}_d + \mathbf{K}\mathbf{e}), \tag{5}$$

where $\mathbf{K}$ is positive definite. Controller 2 added a secondary joint-space task projected into the Jacobian null space:

$$\dot{\mathbf{q}}_{\text{ref},2} = \mathbf{J}^{\dagger}(\dot{\mathbf{h}}_d + \mathbf{K}\mathbf{e}) + \left(\mathbf{I}_4 - \mathbf{J}^{\dagger}\mathbf{J}\right)\mathbf{K}_r\mathbf{e}_q, \tag{6}$$

Here $\mathbf{e}_q = \mathbf{q}_d - \mathbf{q}$, with $\mathbf{q}_d = [0^\circ, 120^\circ, -100^\circ, 0^\circ]^T$ and $\mathbf{K}_r = \text{diag}(0, 1, 0.5, 0)$. Thus, the secondary objective is posture regulation, mainly through the second and third reduced coordinates. With $\mathbf{N} = \mathbf{I}_4 - \mathbf{J}^{\dagger}\mathbf{J}$, the positive sign in Equation (6) gives $\dot{\mathbf{q}}_{\text{sec}} = \mathbf{N}\mathbf{K}_r(\mathbf{q}_d - \mathbf{q})$ and, for constant $\mathbf{q}_d$, $\dot{\mathbf{e}}_{q,\text{sec}} = -\mathbf{N}\mathbf{K}_r\mathbf{e}_q$. This is a projected posture-regulation action; independent global convergence of the complete joint-space error is not claimed [6,7].

When $\mathbf{J}$ has full row rank, $\mathbf{J}\mathbf{J}^{\dagger} = \mathbf{I}_3$ and $\mathbf{J}(\mathbf{I}_4 - \mathbf{J}^{\dagger}\mathbf{J}) = \mathbf{0}$. Therefore, Controllers 1 and 2 have the same ideal primary dynamics, $\dot{\mathbf{e}} = -\mathbf{K}\mathbf{e}$. For $V = \frac{1}{2}\mathbf{e}^T\mathbf{e}$,

$$\dot{V} = -\mathbf{e}^T\mathbf{K}\mathbf{e} < 0, \qquad \mathbf{e} \neq \mathbf{0}, \tag{7}$$

which establishes asymptotic convergence under the stated assumptions.

To reduce abrupt joint commands, Controller 3 replaced linear error feedback with a hyperbolic-tangent term:

$$\dot{\mathbf{q}}_{\text{ref},3} = \mathbf{J}^{\dagger}\left[\dot{\mathbf{h}}_d + \mathbf{K}_1 \tanh(\mathbf{K}_2\mathbf{e})\right]. \tag{8}$$

For positive diagonal $\mathbf{K}_1$ and $\mathbf{K}_2$, full row rank gives $\dot{\mathbf{e}} = -\mathbf{K}_1 \tanh(\mathbf{K}_2\mathbf{e})$ and

$$\dot{V} = -\sum_{i=1}^{3} k_{1,i} e_i \tanh(k_{2,i} e_i) < 0 \quad (\mathbf{e} \neq \mathbf{0}),$$

because $e_i \tanh(k_{2,i} e_i) > 0$ for $e_i \neq 0$. Hence, the ideal continuous-time Cartesian error is asymptotically stable; the discrete-time implementation is evaluated numerically below.

2.3. Simulation Scenarios and Evaluation Criteria

All revised comparisons used the same reduced model, $T_s = 0.1$ s, MATLAB's Moore–Penrose pseudoinverse `pinv(J)` with its default tolerance, and Forward Euler integration, $\mathbf{q}_{k+1} = \mathbf{q}_k + T_s\dot{\mathbf{q}}_k$. For the standardized circular and heart comparisons, $\mathbf{K} = \mathrm{diag}(3, 3, 1)$; Controller 2 used the $\mathbf{q}_d$ and $\mathbf{K}_r$ reported above; and Controller 3 used $\mathbf{K}_1 = \mathbf{I}_3$ and $\mathbf{K}_2 = \mathbf{K}$. Initial configurations were $[0°, 5°, 45°, 0°]^T$ for the circular path and $[45°, 45°, -45°, 0°]^T$ for the heart path.

The circular reference was $\mathbf{h}_d = [0.5\cos(0.2t),\ 0.5\sin(0.2t),\ 0.5]^T$ for $0 \le t \le 40$ s. The figure-eight reference was $[0.3\cos(0.1t),\ 0.3\sin(0.2t),\ 0.6]^T$ for $0 \le t \le 60$ s. The heart reference was $h_{x,d} = [1.5\sin(0.1t) - 0.4\sin(0.3t)]/4$, $h_{y,d} = [1.2\cos(0.1t) - 0.5\cos(0.2t) - 0.1\cos(0.3t) - 0.1\cos(0.4t)]/4$, and $h_{z,d} = 0.6$ for $0 \le t \le 60$ s; analytical derivatives were used for these parametric paths. The former "square" test is more precisely a point-defined piecewise-linear 3D path through $(0.5, 0, 0)$, $(0.4, 0.2, 0.48)$, $(0.4, 0.4, 0.5)$, $(0.45, 0.3, 0.1)$, and $(0.5, 0, 0)$; its corners introduce reference-velocity discontinuities.

Performance was compared using 3D Cartesian RMSE, maximum Cartesian error, peak joint velocity, integrated squared joint-velocity demand $\int \|\dot{\mathbf{q}}\|^2 dt$, minimum singular value $\sigma_{\min}(\mathbf{J})$, and maximum Jacobian condition number $\kappa(\mathbf{J})$. Controllers 1–3 were all evaluated on the same heart-shaped trajectory; the comparison is deterministic and no statistical significance is claimed.The MATLAB implementation used for the standardized controller comparisons and the calculation of the reported performance metrics is provided in Supplementary File S1.

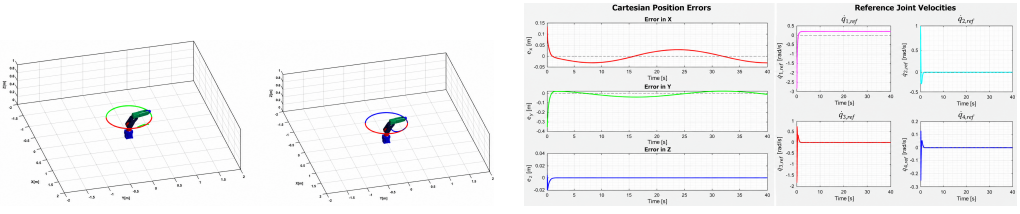
3. Results and Discussion

Table 2 summarizes the standardized simulations. For the circular path, Controllers 1 and 2 produced nearly identical 3D RMSE values (0.00953 and 0.00949 m), while the null-space controller increased peak joint velocity from 0.681 to 1.330 rad/s. The RMSE of the C1–C2 Cartesian position difference was 5.51×10^{-4} m and the maximum Cartesian leakage of the projected null-space term was 5.76×10^{-15}, consistent with $\mathbf{JN} = \mathbf{0}$.

Table 2. Common performance metrics under standardized simulation conditions.

Path	Controller	RMSE (m)	$e_{\max}$ (m)	$\dot{q}_{\max}$ (rad/s)	$\int \|\dot{q}\|^2 dt$	$\sigma_{\min}$	$\kappa_{\max}$
Circular	C1	0.00953	0.09123	0.681	1.713	0.0716	9.07
Circular	C2	0.00949	0.09123	1.330	4.451	0.0534	11.64
Heart	C1	0.03008	0.45231	10.533	24.021	0.0201	33.55
Heart	C2	0.03008	0.45231	10.547	24.114	0.0205	32.88
Heart	C3	0.03089	0.45231	8.171	18.026	0.0201	33.49

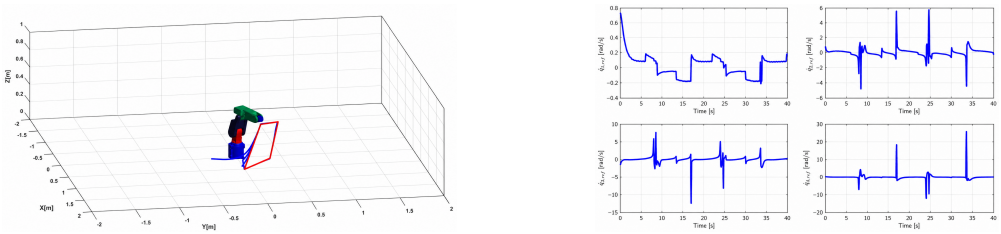
Figure 2 illustrates the circular experiment. The almost unchanged Cartesian error confirms that the null-space term does not directly improve the primary task under the stated rank condition. Its main effect was internal: peak velocity and integrated velocity demand increased, and the maximum condition number changed from 9.07 to 11.64.



(**a**) (**b**)

Figure 2. Circular-trajectory experiment: (**a**) trajectories generated by the two control formulations and (**b**) Cartesian errors and joint-velocity references for Controller 2.

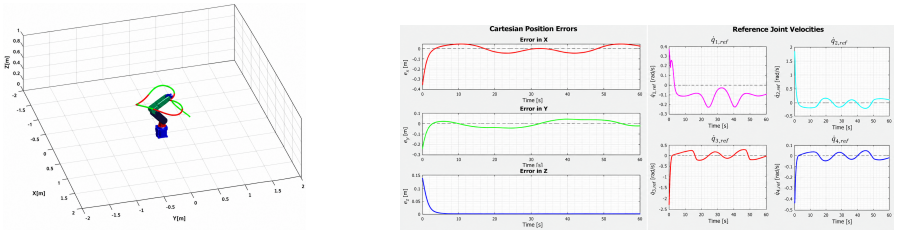
The figure-eight and heart-shaped references were continuous, whereas the point-defined piecewise-linear 3D path contained abrupt changes of direction. As shown in Figure 3, its vertices generated repeated error transients and reference-velocity peaks near 6 rad/s in $q_{2,\mathrm{ref}}$ and 25 rad/s in $q_{4,\mathrm{ref}}$. Extending the horizon from 40 to 120 s did not remove these peaks; therefore, they are attributed primarily to reference discontinuities and pseudoinverse sensitivity rather than to controller design alone.



(**a**) (**b**)

Figure 3. Point-defined piecewise-linear path over a 40-s horizon: (**a**) path followed by the virtual manipulator and (**b**) joint-velocity references.

On the common heart-shaped trajectory (Figure 4), Controller 3 produced a slightly higher RMSE (0.03089 m) than Controllers 1 and 2 (0.03008 m), but reduced peak joint velocity to 8.171 rad/s and integrated velocity demand to 18.026 rad^2/s, compared with 10.533 rad/s and 24.021 rad^2/s for Controller 1. Thus, nonlinear attenuation reduced joint-space demand without being interpreted as a direct improvement in Cartesian accuracy.



(**a**) (**b**)

Figure 4. Heart-shaped trajectory with nonlinear attenuation: (**a**) followed path and (**b**) Cartesian errors and joint-velocity response.

Overall, position error alone is insufficient for controller comparison. According to the RV-2FR manufacturer specifications, nominal maximum speeds are 2.62 rad/s for J2 and 7.85 rad/s for J5; thus, the approximately 6 and 25 rad/s commands observed in the discontinuous path are not physically executable [11]. Joint-position limits were not enforced in the kinematic simulation: for example, the heart test reached approximately −259.2° in the base coordinate and 146.9° in the second coordinate, exceeding the nominal J1 and J2 ranges. These outputs are therefore numerical stress-test results, not commands proposed for direct deployment.

The study remains limited to an unconstrained reduced kinematic model; actuator dynamics, torque limits, friction, backlash, communication delays, sensor noise, and mechanical flexibility were not represented. Singularity-robust inversion, smooth corner interpolation, explicit joint/velocity constraints, and experimental validation are required before physical implementation [8,9].

4. Conclusions

A reduced 4-DOF model of the Mitsubishi MELFA RV-2FRV robotic manipulator was developed in MATLAB to evaluate and compare three kinematic control approaches: primary Cartesian control, null-space secondary-task control, and nonlinear velocity at-

tenuation. The virtual model integrated the manipulator geometry, forward kinematics, Jacobian-based velocity mapping, and different reference trajectories, providing a flexible environment for assessing controller performance before implementation on the physical platform.

Under identical circular-path conditions, Controllers 1 and 2 produced nearly equal Cartesian RMSE (0.00953 and 0.00949 m), while the null-space posture task increased joint-velocity demand. On the common heart path, nonlinear attenuation reduced peak velocity from 10.533 to 8.171 rad/s and integrated demand from 24.021 to 18.026 rad^2/s, with only a small RMSE increase.

The discontinuous point-defined path generated non-executable velocity peaks and joint excursions, showing the need for smooth references and explicit physical constraints. The framework remains a comparative kinematic simulation; future work should include joint/velocity/acceleration limits, singularity-robust inversion, full dynamics, and experimental validation.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/engproc2026158022/s1, Supplementary File S1: `Unified_Controller_Comparison_Reviewer.m`, containing the standardized MATLAB simulations and metric calculations used in the comparative analysis reported in this paper.

Author Contributions: Conceptualization, J.G.-M. and M.X.L.; methodology, J.G.-M.; software, L.A.F.A. and M.X.L.; validation, M.A.V., J.G.-M. and M.X.L.; formal analysis, M.X.L.; investigation, J.G.-M.; resources, J.G.-M.; data curation, M.X.L.; writing—original draft preparation, M.A.V.; writing—review and editing, J.G.-M. and M.A.V. and M.X.L.; visualization, L.A.F.A. and J.G.-M.; supervision, M.X.L. and J.G.-M.; project administration, M.X.L.; funding acquisition, J.G.-M. and L.A.F.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data supporting the findings of this study are available within the article and Supplementary File S1.

Acknowledgments: The authors gratefully acknowledge the institutional support provided by the Instituto Superior Tecnológico Martha Bucaram de Roldós and the collaboration of researchers from the Universidad Técnica de Ambato. The results presented in this study were obtained within the research project entitled "Trajectory Description for Forward Kinematics Based on the Implementation of a Robotic Arm", approved under project code PID-MCT-TI-010.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Holmes, N.; Katavich, L.; Xu, X. Retrofitting legacy systems for Industry 4.0 via OPC UA and distributed control. *Manuf. Lett.* **2025**, *44*, 1337–1348. [CrossRef]
2. Andaluz, V.H.; Pérez, J.A.; Carvajal, C.P.; Ortiz, J.S. Virtual environment for teaching and learning robotics applied to industrial processes. In *Lecture Notes in Computer Science*; Springer International Publishing: Cham, Switzerland, 2019; pp. 442–455. [CrossRef]
3. Wolf, A.; Romeder-Finger, S.; Széll, K.; Galambos, P. Towards robotic laboratory automation plug & play: Survey and concept proposal on teaching-free robot integration with the LAPP digital twin. *SLAS Technol.* **2023**, *28*, 82–88. [CrossRef]
4. Wan, K.; Li, C.; Lo, F.-S.; Zheng, P. A virtual reality-based immersive teleoperation system for remote human-robot collaborative manufacturing. *Manuf. Lett.* **2024**, *41*, 43–50. [CrossRef]
5. Talli, A.; Meti, V.K.V. Design, simulation, and analysis of a 6-axis robot using robot visualization software. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *872*, 012040. [CrossRef]
6. Liègeois, A. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Trans. Syst. Man Cybern.* **1977**, *7*, 868–871. [CrossRef]
7. Nakamura, Y.; Hanafusa, H.; Yoshikawa, T. Task-priority based redundancy control of robot manipulators. *Int. J. Robot. Res.* **1987**, *6*, 3–15. [CrossRef]

8. Chiaverini, S. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Trans. Robot. Autom.* **1997**, *13*, 398–410. [CrossRef]
9. Wampler, C.W. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Trans. Syst. Man Cybern.* **1986**, *16*, 93–101. [CrossRef]
10. Abaas, T.F.; Khleif, A.A.; Abbood, M.Q. Inverse kinematics analysis and simulation of a 5 DOF robotic arm using MATLAB. *Al-Khwarizmi Eng. J.* **2020**, *16*, 1–10. [CrossRef]
11. Mitsubishi Electric. *RV-FR Series Standard Specifications Manual*; BFP-A3470-AA; Mitsubishi Electric Corporation: Tokyo, Japan, 2024. Available online: https://www.fapro.com.tw/DB/download/mitsubishi_ROBOT/Manual/RV-FR/English/bfp-a3470y.pdf (accessed on 22 September 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.